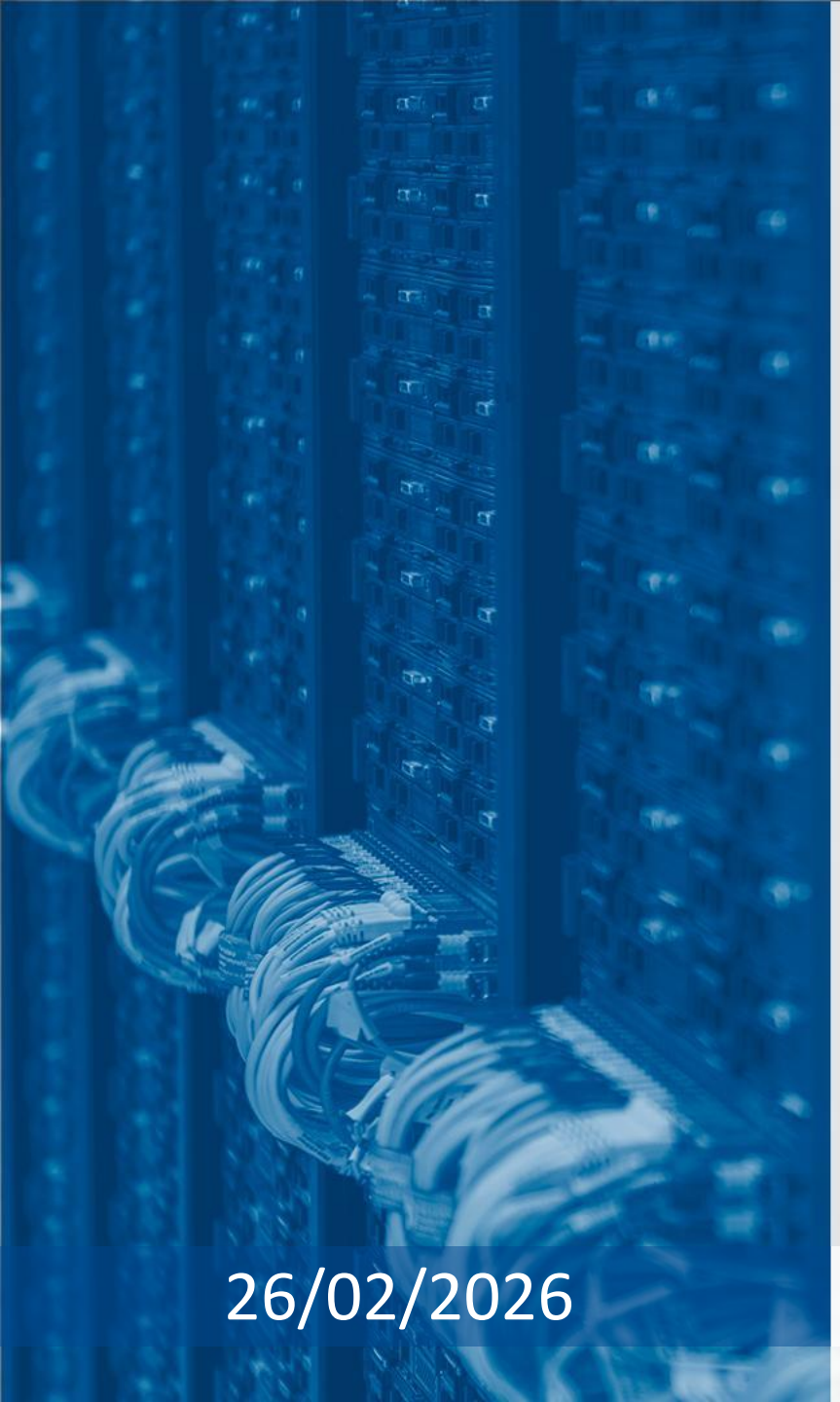


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Providentia
Training Session 2

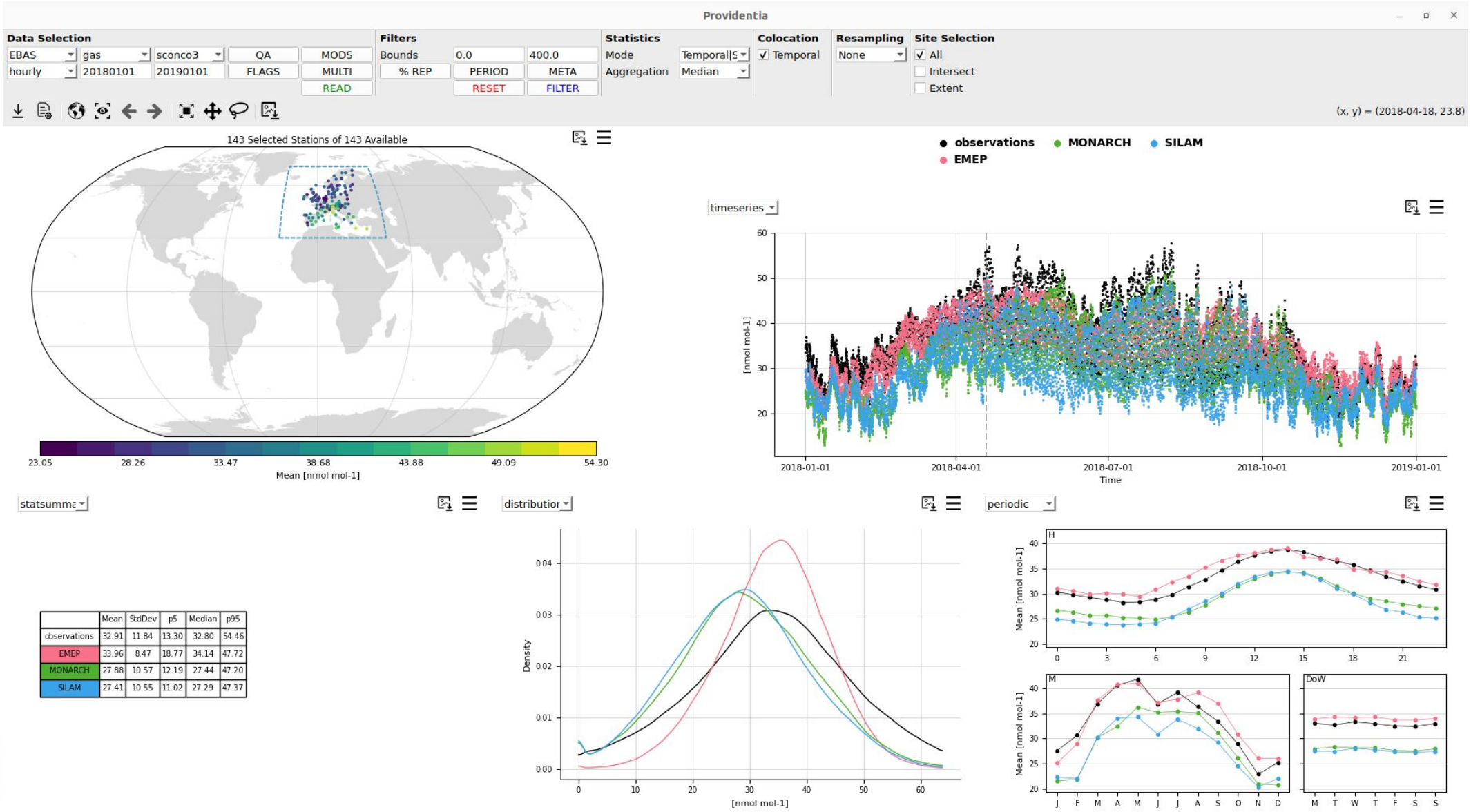
26/02/2026

Paula Serrano | Alba Vilanova | Dene Bowdalo

Dashboard

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Dashboard

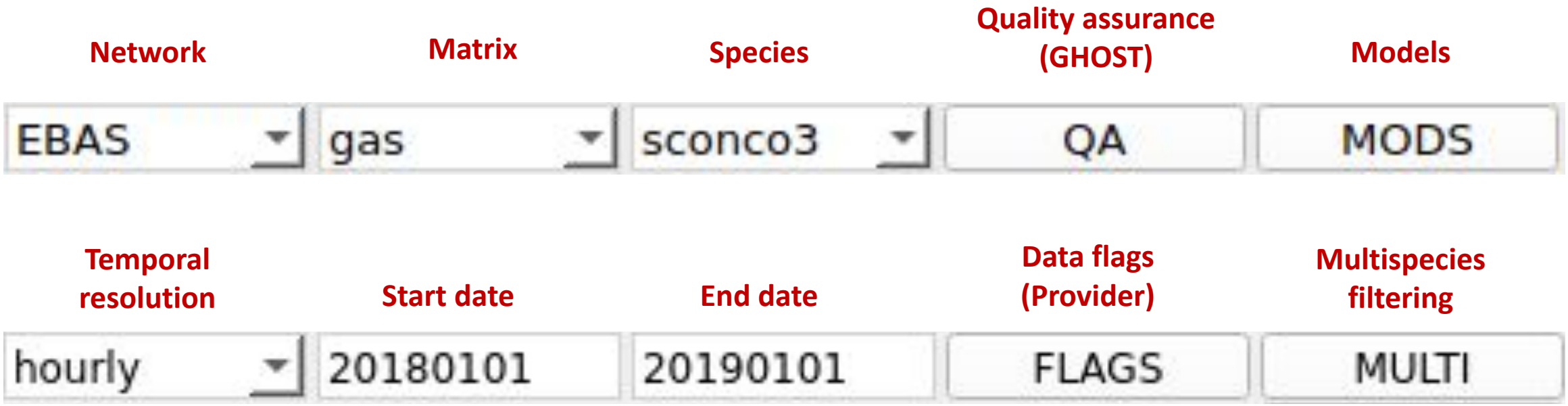


	Mean	StdDev	p5	Median	p95
observations	32.91	11.84	13.30	32.80	54.46
EMEP	33.96	8.47	18.77	34.14	47.72
MONARCH	27.88	10.57	12.19	27.44	47.20
SILAM	27.41	10.55	11.02	27.29	47.37

Main menu

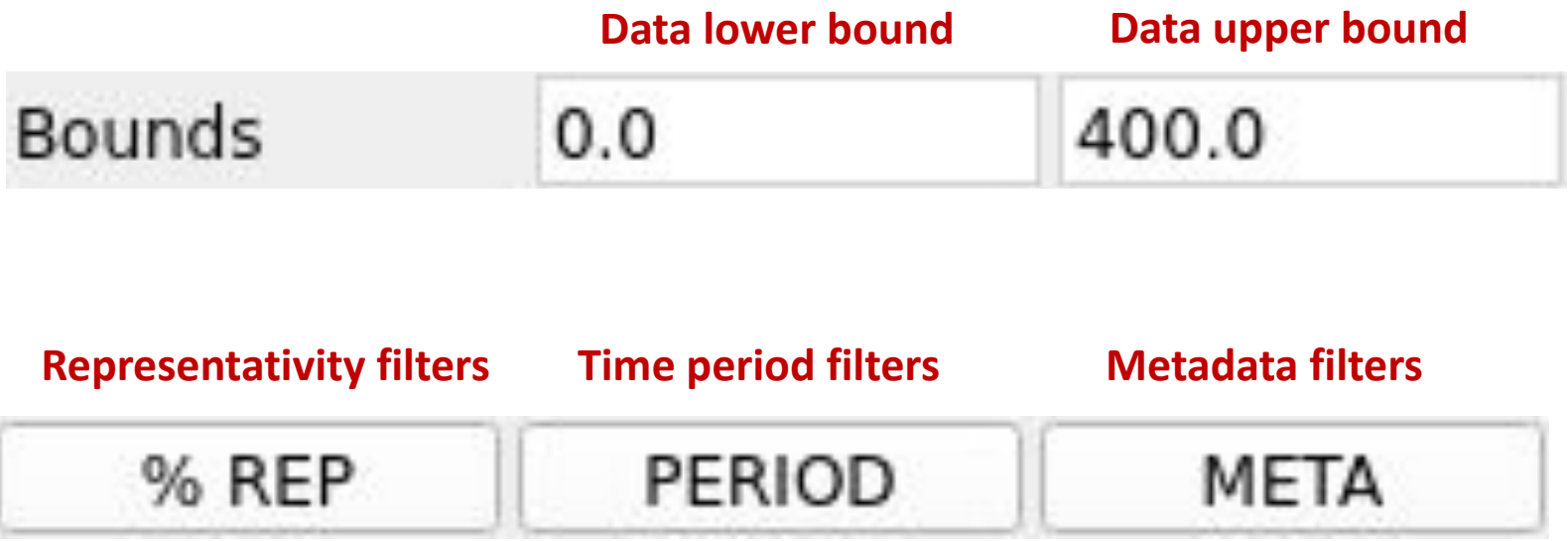

Data Selection
EBAS
gas
sconco3
QA
MODS
hourly
20180101
20190101
FLAGS
MULTI
READ
Filters
Bounds
0.0
400.0
% REP
PERIOD
META
RESET
FILTER
Statistics
Mode
Temporal|S
Aggregation
Median
Colocation
Temporal
Resampling
None
Site Selection
All
Intersect
Extent

Data selection menu

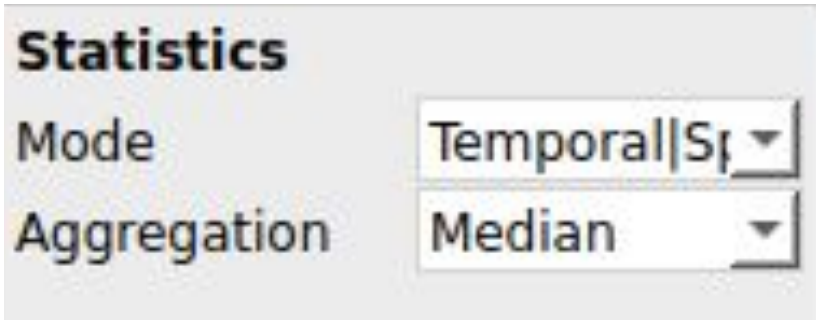


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Filters menu



Statistics menu



Statistic mode

Statistic aggregation

Colocation menu

Colocate models vs. observations, removing temporal gaps

Resampling menu

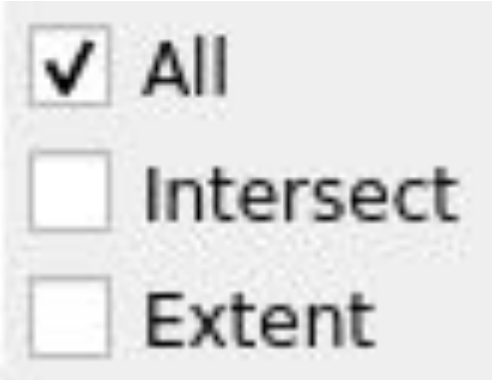
Temporal resolution to resample your data to (always lower than the selected resolution)

Stations selection menu

Select all stations

Select intersecting stations within all model domains

Select stations on current map view



Icons menu

Upload configuration files

Set original view

Forward

Pan

Save canvas

Download data and configuration files

Set world view

Back

Zoom to rectangle

Lasso

Interactive features

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Navigation tips

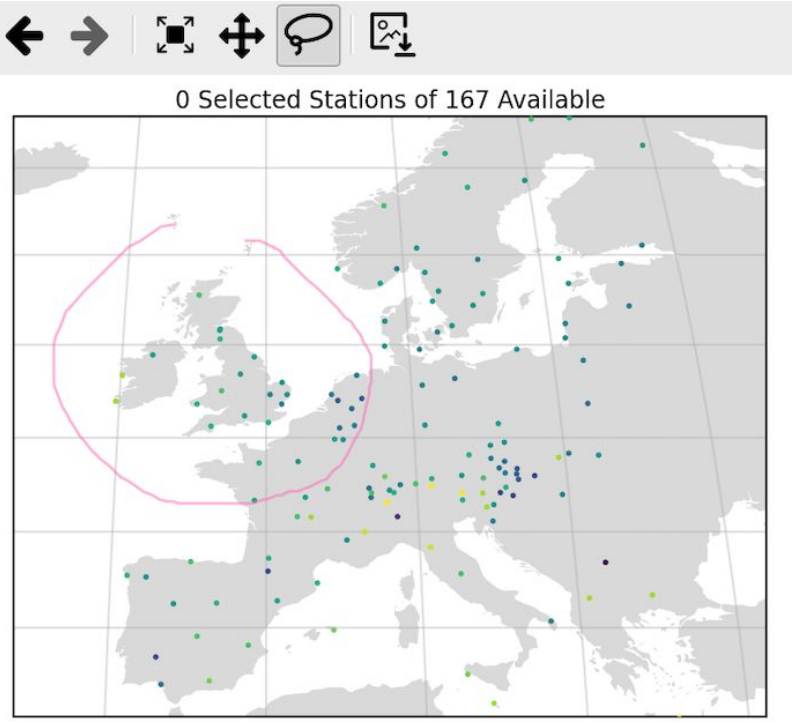
Individual stations can be selected by left clicking on them on the map.

Multiple stations can be selected by activating the lasso, and drawing it round all desired stations.

To unselect stations simply click on an empty space on the map.

When multiple stations are selected, to add or remove any individual station right click on the given station.

To quickly zoom on the map, the mouse or trackpad scroll wheel can be used.

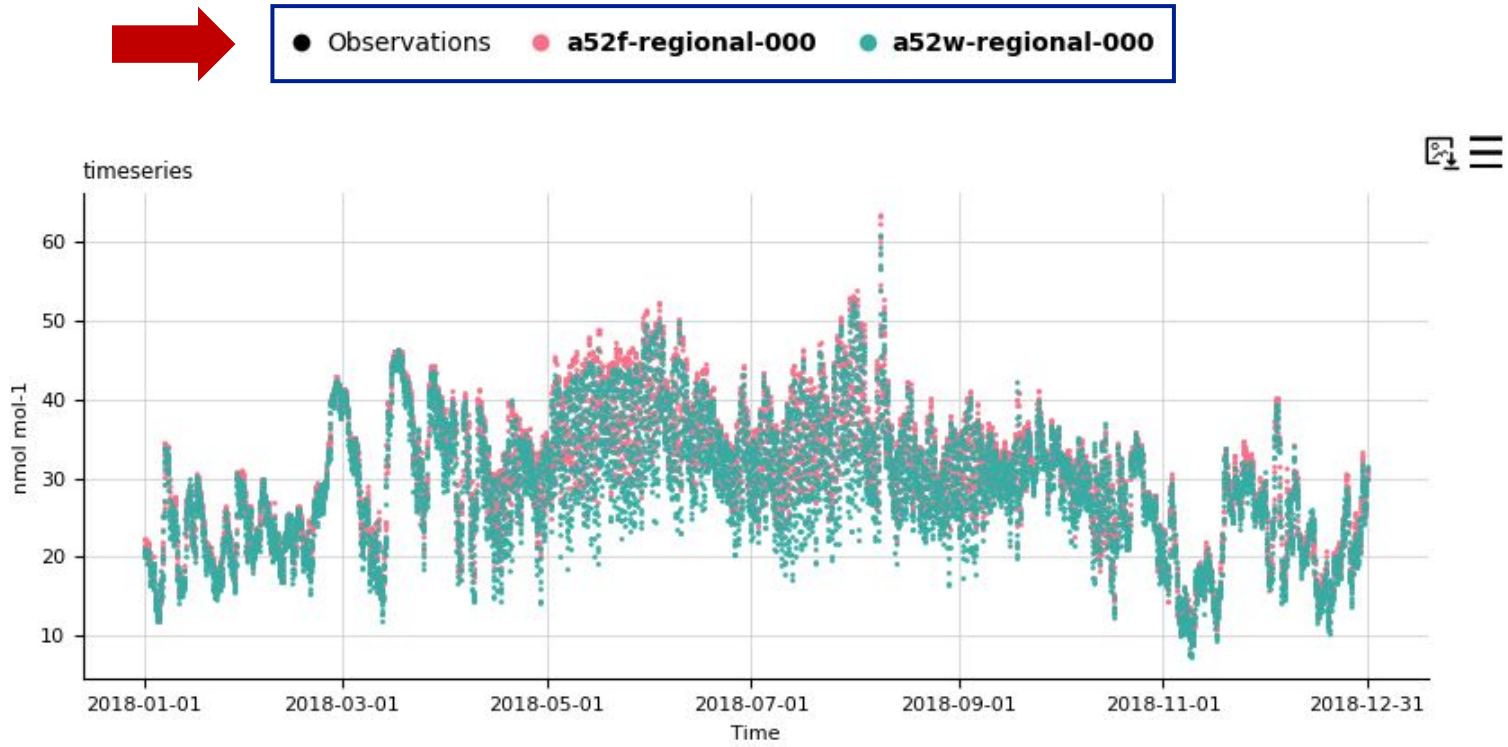


Example of lasso selection

Legend picking

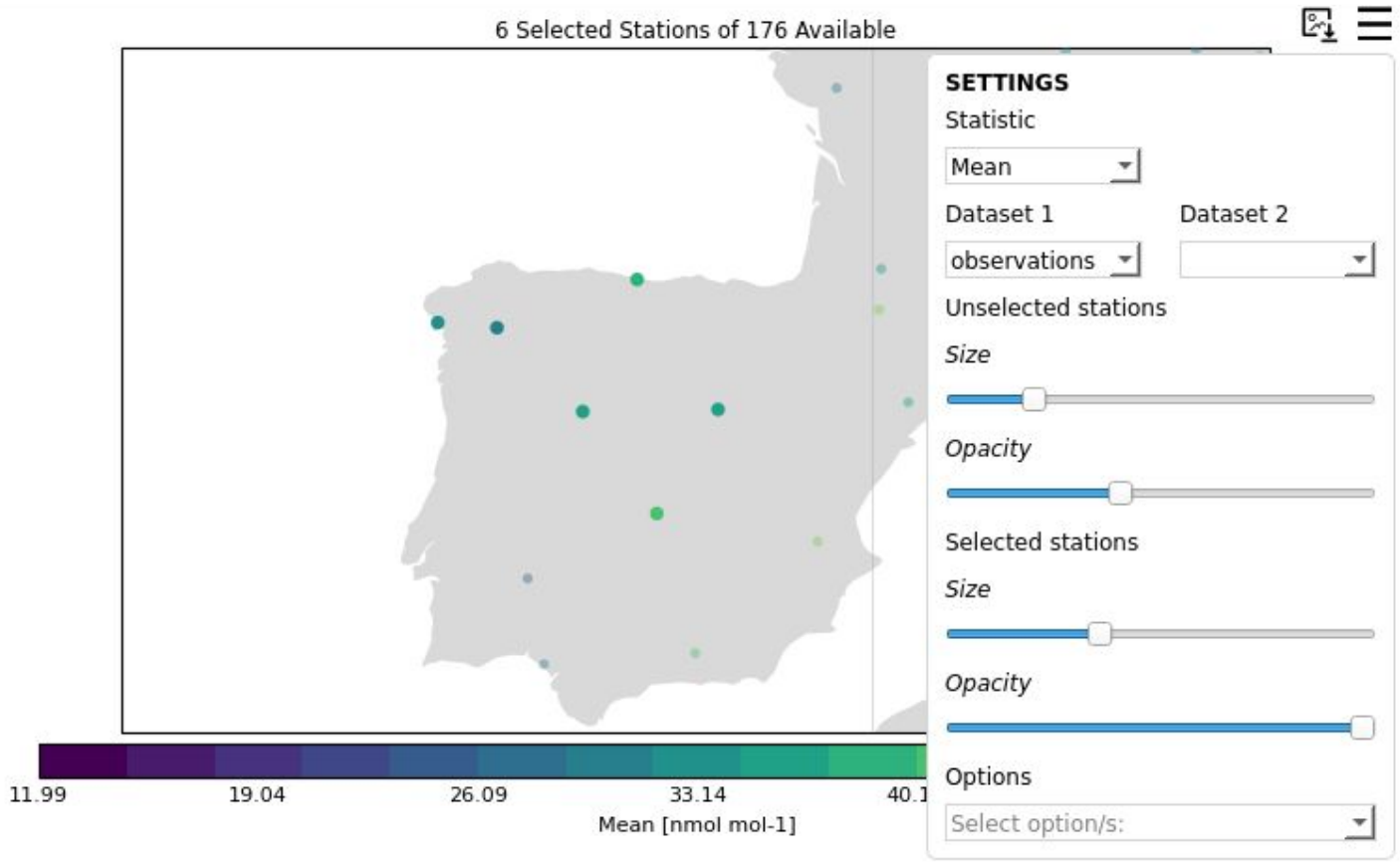
Clicking on the legend labels will remove or add data to each of the plots

Bold = Visible
Roman = Invisible



Customise the plot style

The style of the plots can be edited by clicking on the **burger menus** and changing the settings.



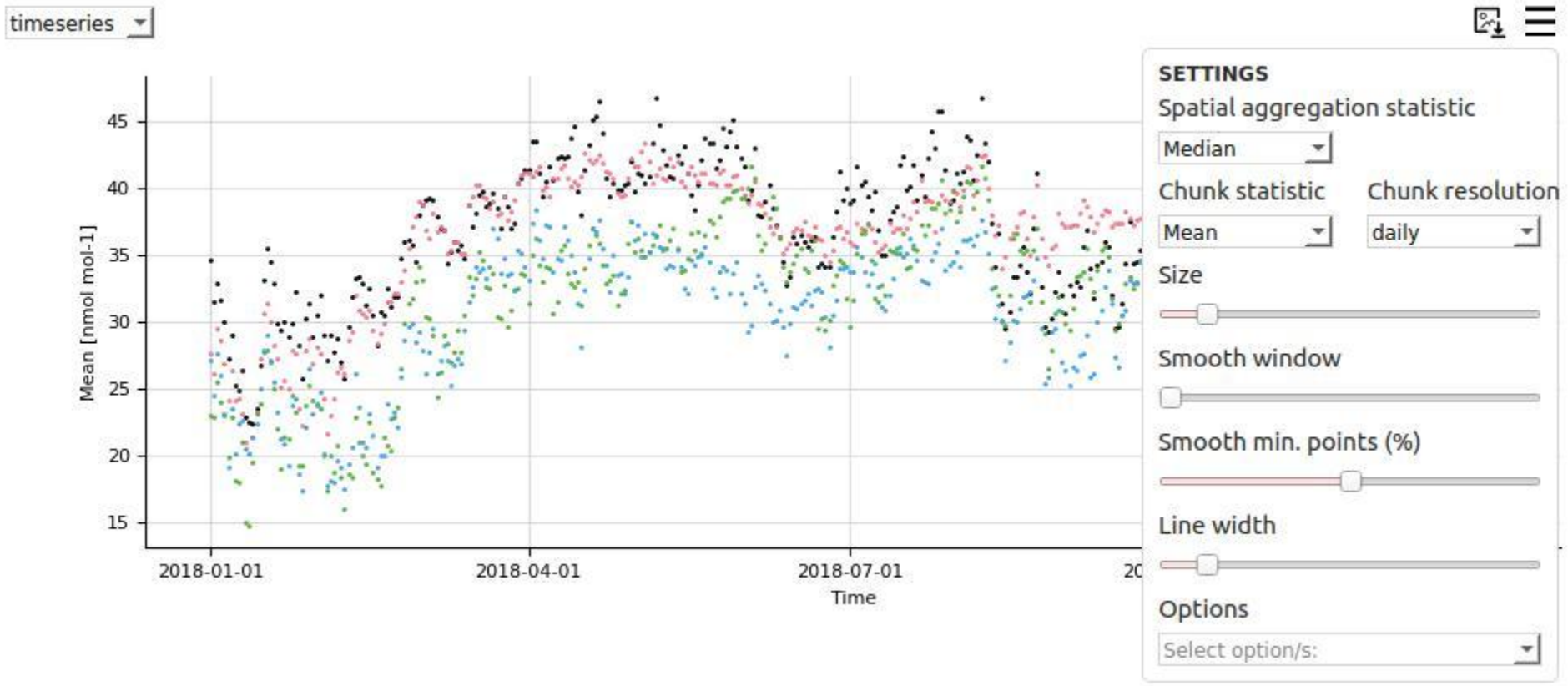
Choosing the statistics

The statistics in the statsummary can be updated from the burger menu.

	Mean	StdDev	p5				
Observations	34.01	13.98	10.96				
cams61_camsra_ph2-eu-000	30.21	13.27	7.40				
cams61_chimere_ph2-eu-000	36.24	9.37	20.07	36.76	50.57	100.98	79.17
cams61_dehm_ph2-eu-000	42.40	10.27	24.92	42.74	58.55	117.41	79.17

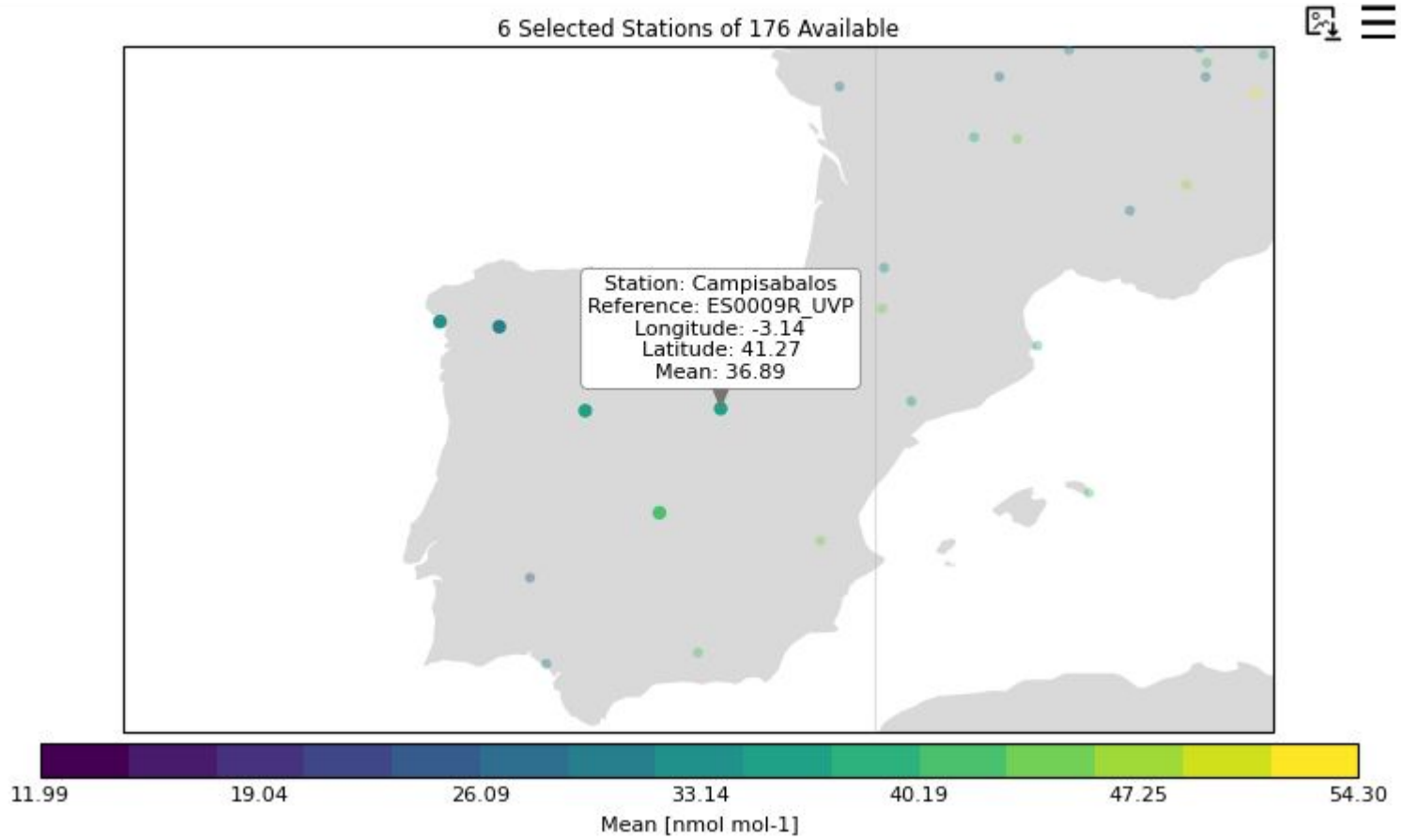
Choosing the statistics

The statistics in other plots can also be updated from the burger menus.



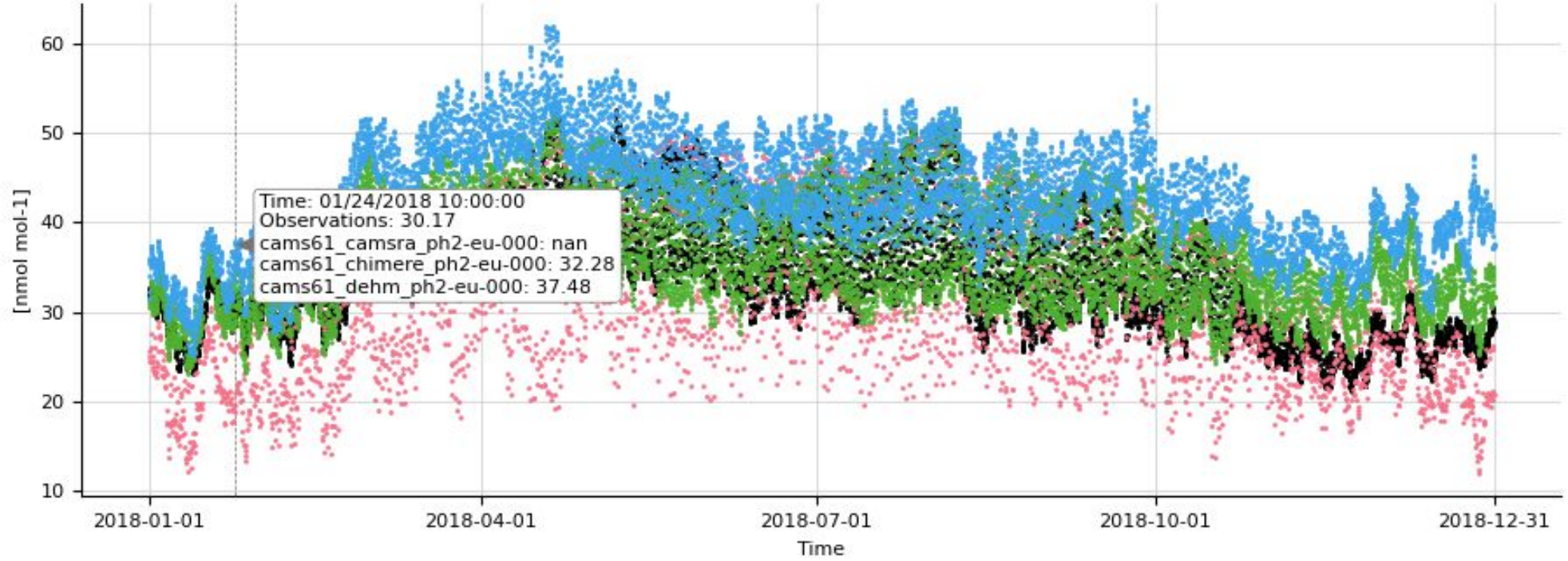
Information on hover

We can see the stations details and data by hovering on the map.



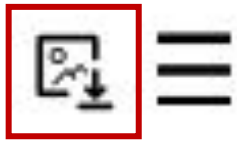
Information on hover

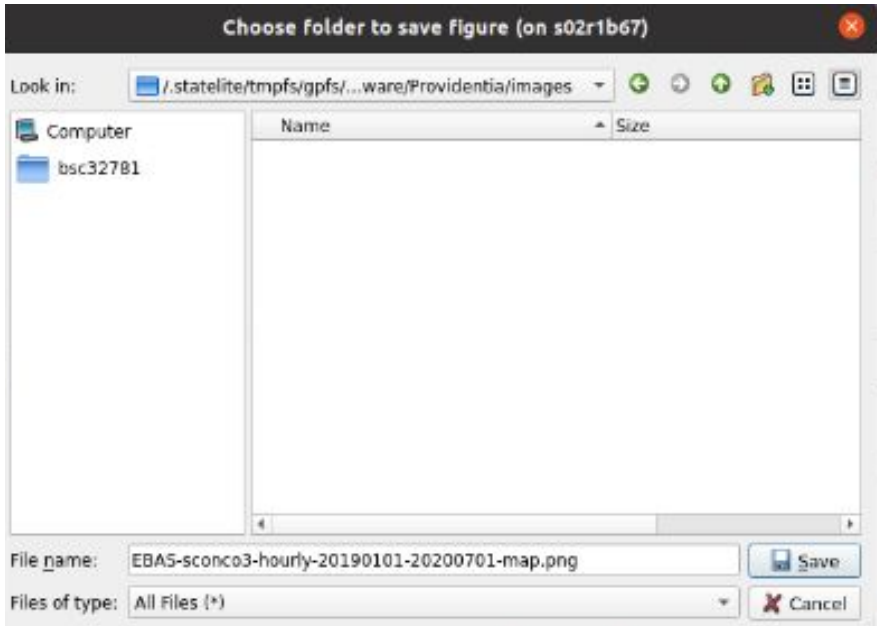
We can also check the values of each dataset by hovering on the other plot types.

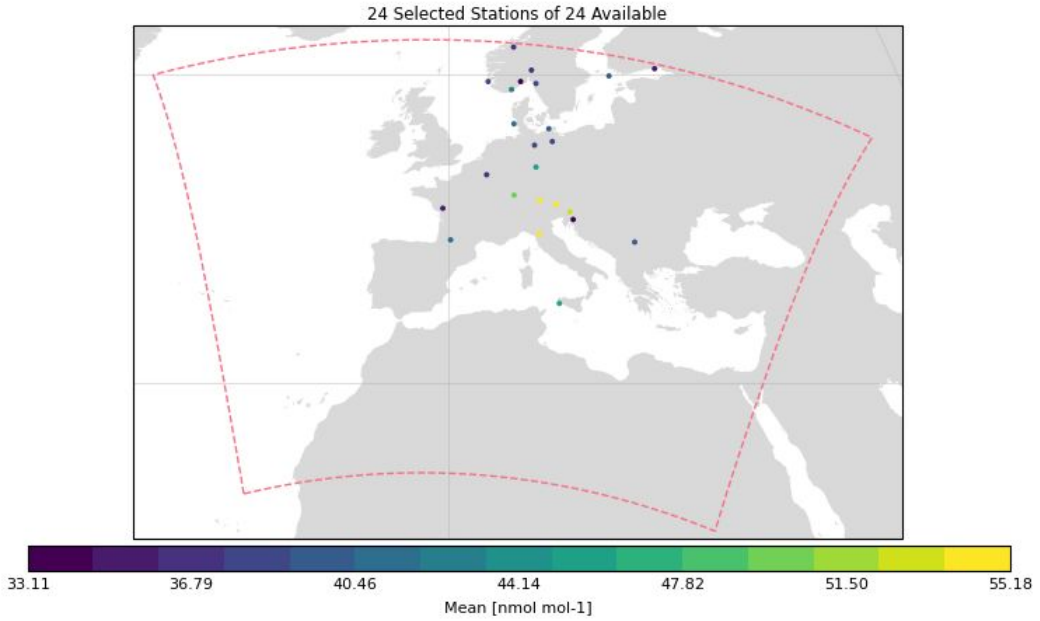


Saving plot figures

Saving plot figures is now possible by clicking on the image icons next to the burger menus

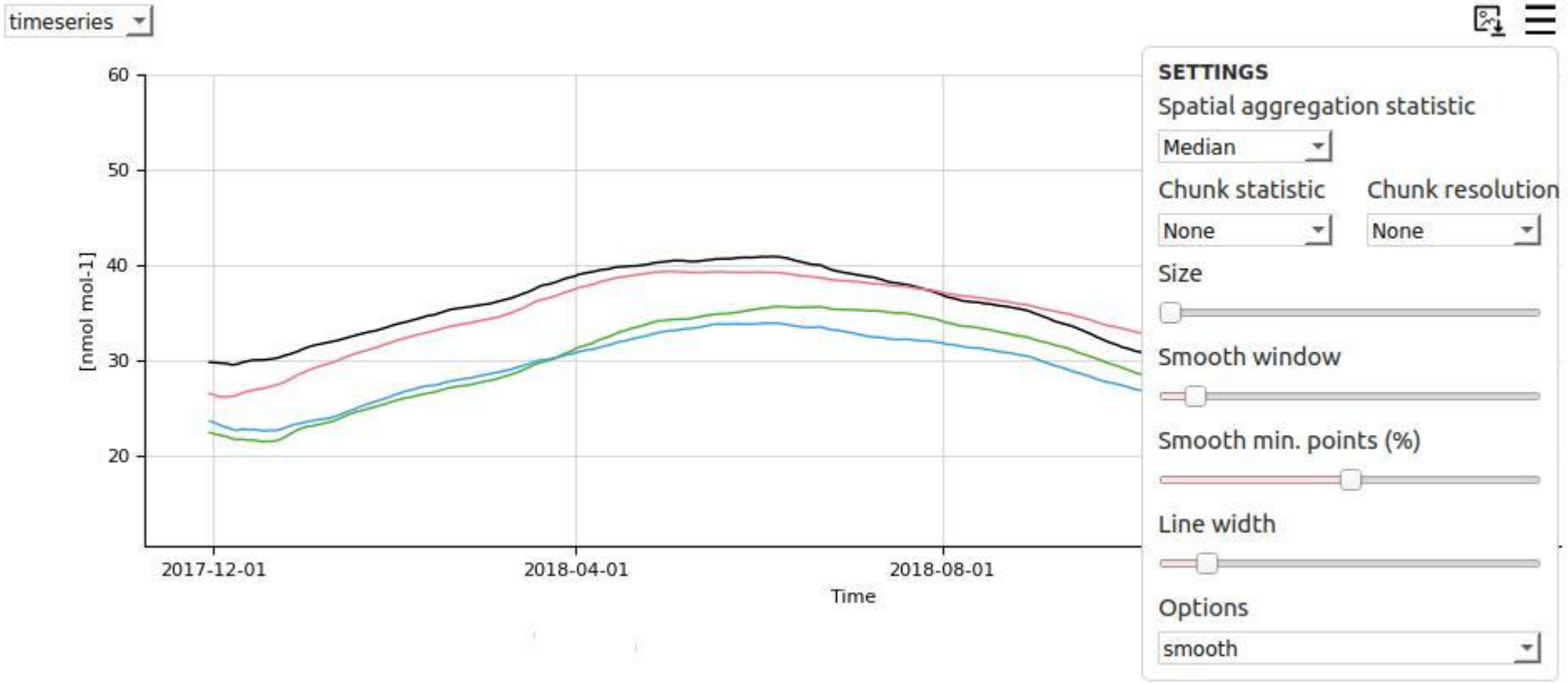






Smoothing

It is possible to add a smooth line in the timeseries plot by editing the smooth window, the data points can be then hidden by reducing their size to 0.



Dashboard specific configuration options

Variable	Purpose
active_dashboard_plots	Defines the plots that will be active in the dashboard once it is launched (e.g. timeseries, periodic-violin, scatter, distribution).

Advanced features

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Filtering

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Metadata filtering

The most common type of data filtering is by metadata, e.g. for **country**, **longitude**, **altitude** etc.

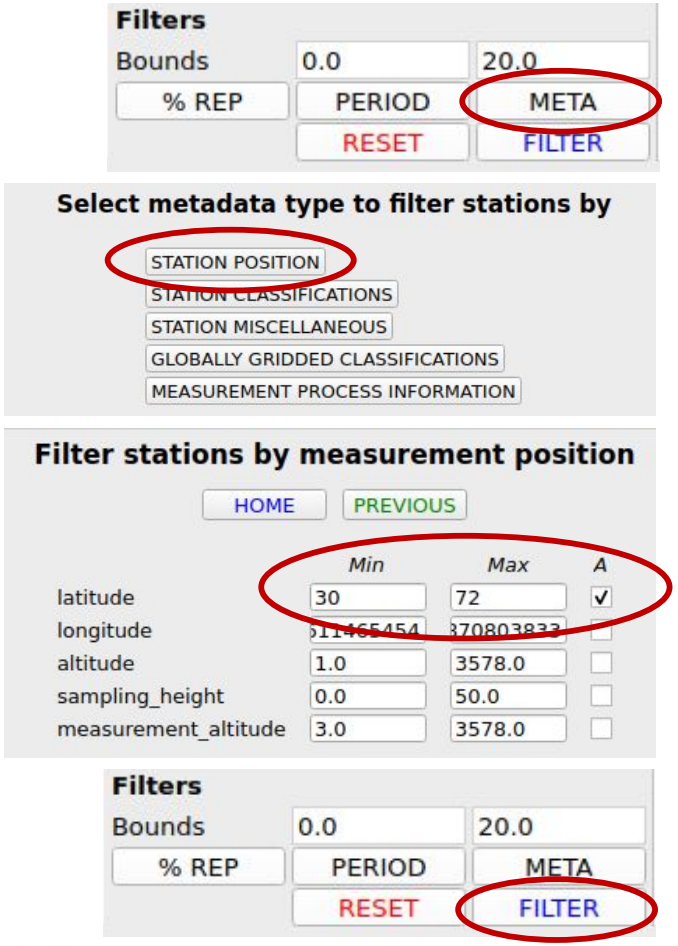
Metadata can either by **numerical** or **text** and the method for filtering for these slightly varies, as will be described.

One common question is what metadata available variables exist for the loaded data that can be filtered by. These can be seen empirically by looking through the sub-menus under the **META** button on the dashboard, with all available variables are organised into 5 categories: **station position**, **station classifications**, **station miscellaneous**, **globally gridded classifications** and **measurement process information**.

Reference:
https://providentia.readthedocs.io/en/latest/Metadata-fields.html

Metadata filtering: Numerical data

Dashboard



Configuration file

latitude = 30, 72

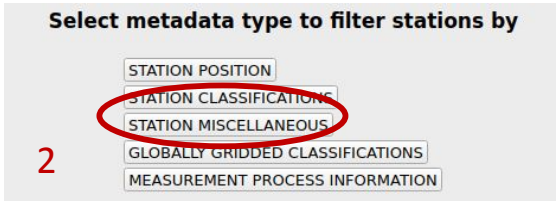
Library

provi.filter('latitude', lower=30, upper=72)

Reference:
https://providentia.readthedocs.io/en/latest/Filtering.html#numerical-metadata

Metadata filtering: Text metadata

Dashboard



Configuration file

```
country = keep: Austria
country = remove: Germany, France
```

Library

```
provi.filter("country", keep="Austria")
provi.filter("country", remove=["Germany", "France"])
```

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

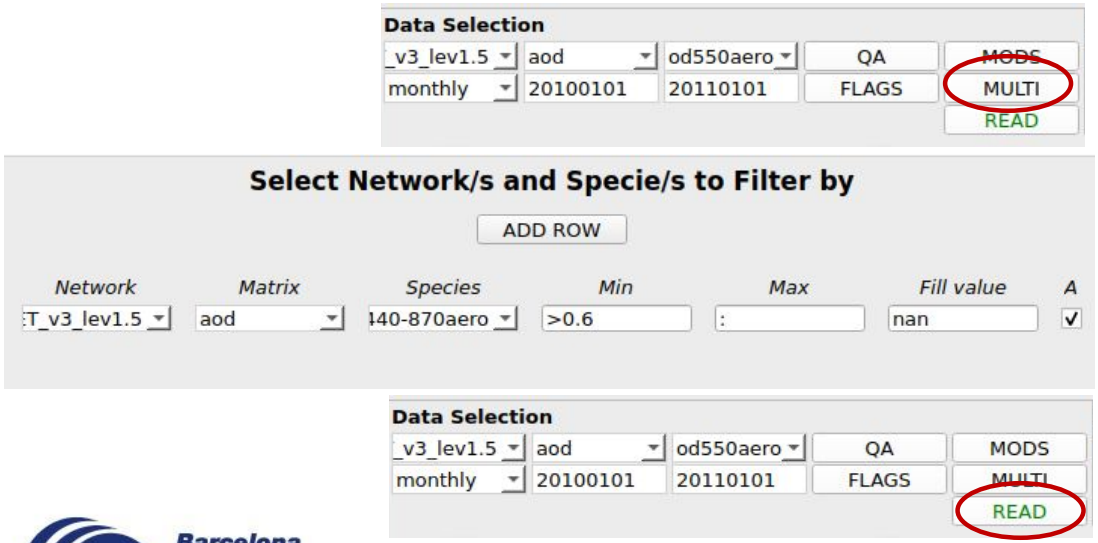
Reference:
https://providentia.readthedocs.io/en/latest/Filtering.html#text-metadata

Multispecies filtering

Multispecies filtering refers to the ability to **filter loaded species data by the values of another species**.

For example, when performing investigations of dust in the atmosphere it is a common practice to filter the AOD by the Angstrom exponent, to isolate values associated with dust.

Dashboard



Configuration file

```
network = AERONET_v3_lev1.5
species = od550aero
filter_species = AERONET_v3_lev1.5:ae440-870aero (>0.6, :, nan)
spatial_colocation = True
```

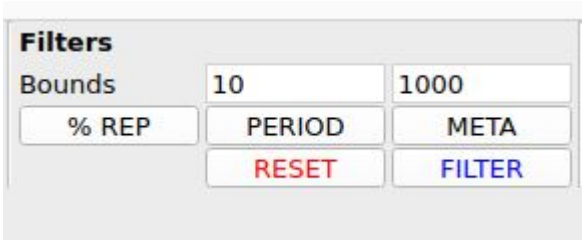
Reference:
https://providentia.readthedocs.io/en/latest/Filtering.html#multispecies-filtering

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Bounds

Often it is desired to remove values which exceed certain extreme bounds, as it is known that data should appear at such extremes. These bounds will be **by default active in Providentia**, with extreme bounds associated with a given species taken from definitions in GHOST.

Dashboard




```
'extreme_lower_limit':0.0, 'extreme_upper_limit':400.0,
'extreme_lower_limit':0.0, 'extreme_upper_limit':1200.0,
'extreme_lower_limit':0.0, 'extreme_upper_limit':600.0,
'extreme_lower_limit':0.0, 'extreme_upper_limit':3000.0,
'extreme_lower_limit':0.0, 'extreme_upper_limit':30000.0,
'extreme_lower_limit':0.0, 'extreme_upper_limit':50000.0,
```

Configuration file

```
lower_bound = 10
upper_bound = 1000
```

Link to GHOST Standards

Reference:

https://providentia.readthedocs.io/en/latest/Filtering.html#bounds

Representativity

Observations often have missing data. If you compare them directly with complete model data, the results can be biased. Representativity filtering helps by only using time periods that have enough data.

You can either apply the representativity filter for an **entire time range** (**all_representativity_percent**) or by **periods**, which are: **day, month** and **year.**

As you change the temporal resolution of the data to be coarser, some of these fields will become unavailable. For example when using monthly data, you can not filter by daily representativity.

Reference: https://providentia.readthedocs.io/en/latest/Filtering.html#available-filters

Available representativity fields

By default they are 0:

Native representativity filters

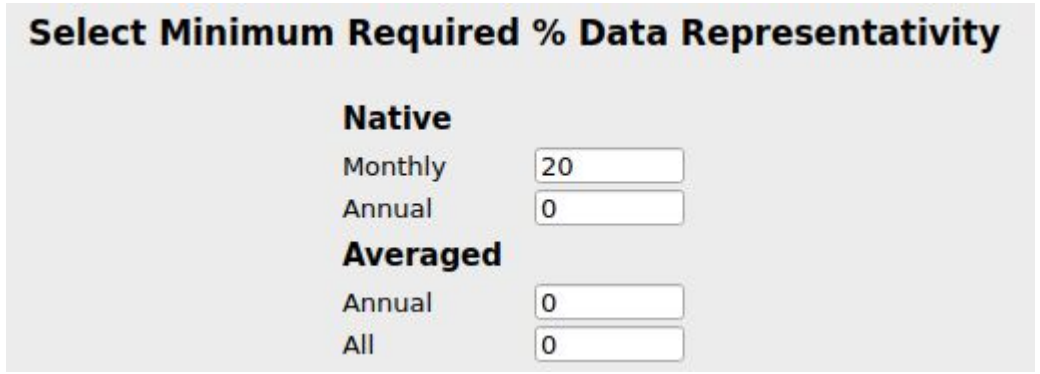
hourly_native_representativity_percent = 0
daily_native_representativity_percent = 0
monthly_native_representativity_percent = 0
annual_native_representativity_percent = 0

Averaged representativity filters

daily_representativity_percent = 0
monthly_representativity_percent = 0
annual_representativity_percent = 0
all_representativity_percent = 0

Representativity

Dashboard



Configuration file

monthly_native_representativity_percent = 20

Library

provi.filter('monthly_native_representativity_percent', limit=20)

Periods

It is often desired to select or remove certain periods over a time range, for example just keep daytime data or remove summertime data.

Providentia gives an easy way to filter in such a way using the **period** variable when using GHOST data.

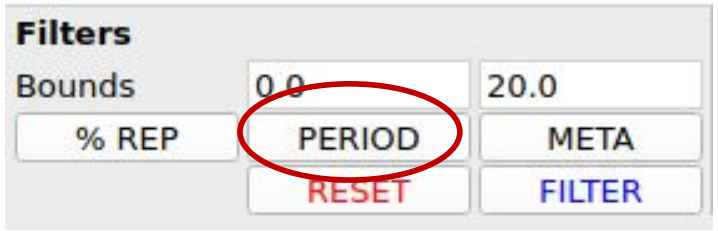
The available period fields that that can be selected are: **Daytime**, **Nighttime**, **Weekday**, **Weekend**, **Spring**, **Summer**, **Autumn** and **Winter**.

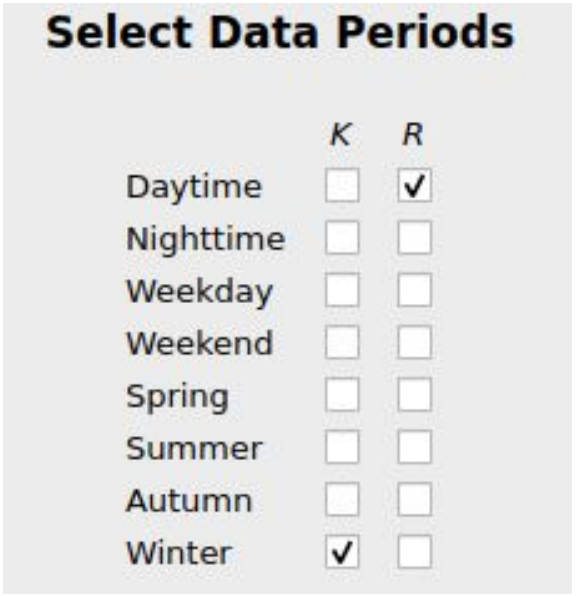
Some fields depend on the current time resolution of the data. For example, you cannot select daytime values when monthly data is loaded.

Reference: https://providentia.readthedocs.io/en/latest/Filtering.html#available-filters

Periods

Dashboard





Configuration file

period = keep: Winter
period = remove: Daytime

Library

provi.filter('period', keep='Winter', remove='Daytime')

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

GHOST QA

GHOST QA refers to quality assurance flags which come from checks designed in GHOST to inform on the quality of data, and can be used to filter measurements.

These can be set via the configuration file either by code or name as follows:

qa = 0, 1, 2

qa = Missing Measurement, Infinite Value, Negative Measurement

QA flags can also be added to, or subtracted from as follows (again, either code or name):

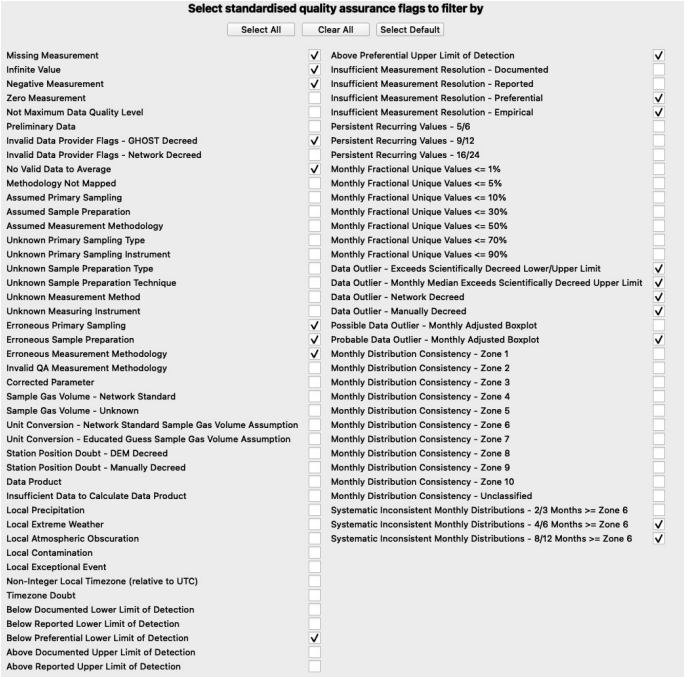
add_qa = Missing Measurement, Negative Measurement

subtract_qa = 0, 2

On the dashboard the GHOST QA flags can be set interactively via the **QA** button.

See here for specific GHOST QA flag definitions and codes.

QA flags



Network QA

Network QA refers to standardised quality assurance flags which come from networks processed in GHOST which inform on the quality of data, and can be used to filter measurements.

These can be set via the configuration file either by code or name as follows:

flags = 0, 1, 2

flags = Valid Data, Preliminary Data, Missing Data

Network flags can also be added to, or subtracted from as follows (again, either code or name):

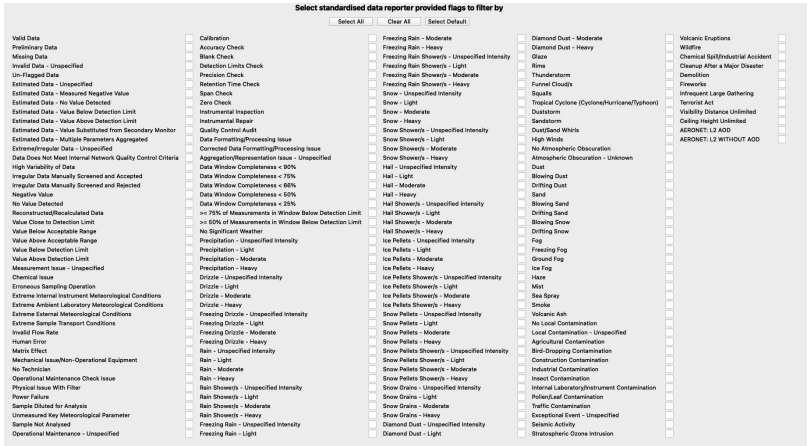
add_flags = Valid Data, Missing Data

subtract_flags = 0, 2

On the dashboard the network flags can be set interactively via the **FLAGS** button.

See here for specific Network QA flag definitions and codes.

Network flags



Calibration

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Calibration factor

When a model has a known bias and you want to correct it, you can apply what we call the calibration factor by defining it in your configuration file.

- To add:

```
calibration_factor = +10
```

- To multiply:

```
calibration_factor = *10
```

- To subtract:

```
calibration_factor = -10
```

- To divide:

```
calibration_factor = /10
```

Calibration factor for multiple models/species

The calibration factor can also be defined independently for different models:

```
calibration_factor = a54s-regional-000 (*0.62), a4xf-regional-000 (*0.51)
```

It can also be defined independently for different species:

```
network = EEA_AQ_eReporting
species = pm2p5, pm10
calibration_factor = a54s-regional-000 (*0.62, *0.4),
a4xf-regional-000 (*0.52, *0.9)
```

Colocation

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Spatial colocation

When loading more than one species (using **multispecies filtering** or in **multispecies plots** in the report and library modes) you may want to ensure that the available stations measure **data for all species** that are to be loaded. To do this, we need to activate spatial colocation.

After activating spatial colocation, any stations that do not have valid data for any of the loaded species are dropped.

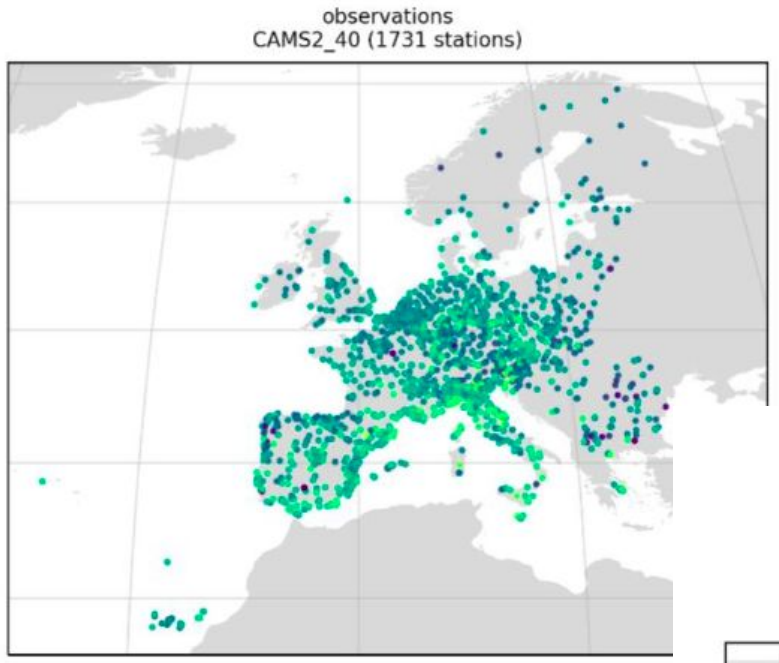
Spatial colocation can be set in the configuration file by setting a boolean as follows (default: True):

```
spatial_colocation = False
```

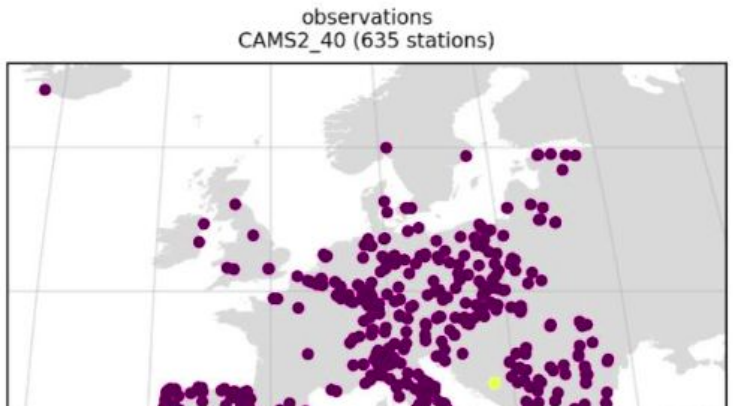

Reference:

https://providentia.readthedocs.io/en/latest/Colocation.html

Surface O_3 (mean)
1731 stations

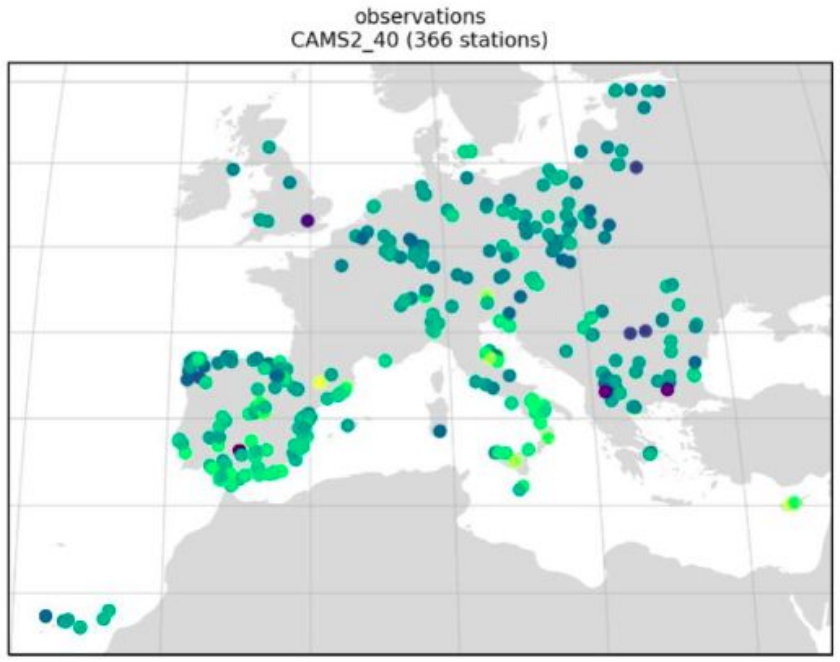


Surface CO (mean)
635 stations

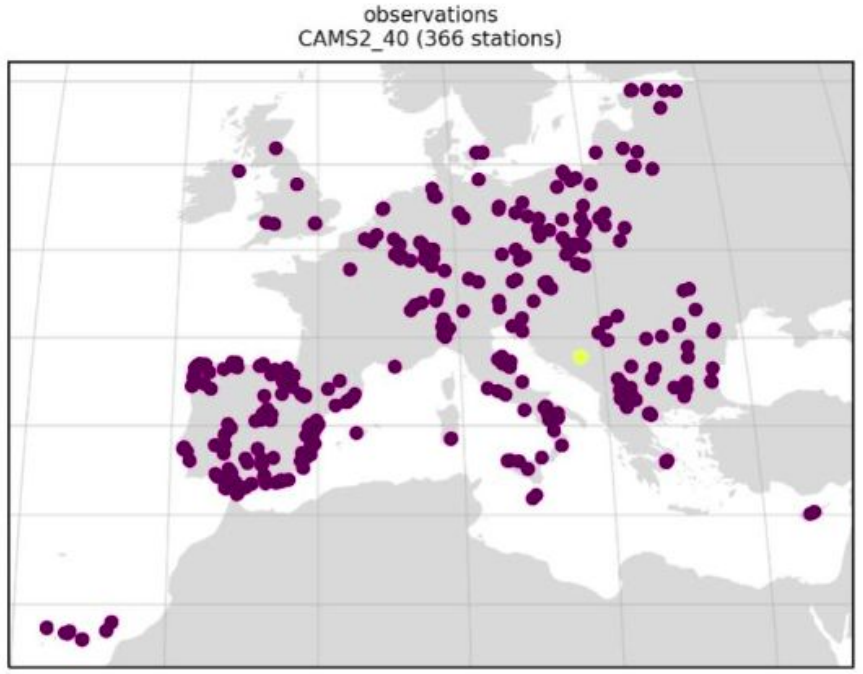


Without spatial colocation

Surface O_3 (mean)
366 stations



Surface CO (mean)
366 stations



With spatial colocation

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Temporal colocation

Temporal colocation is used to **temporally pair observations and model data**, with any missing measurements in either the observational or model array, imposing missing measurements on the other.

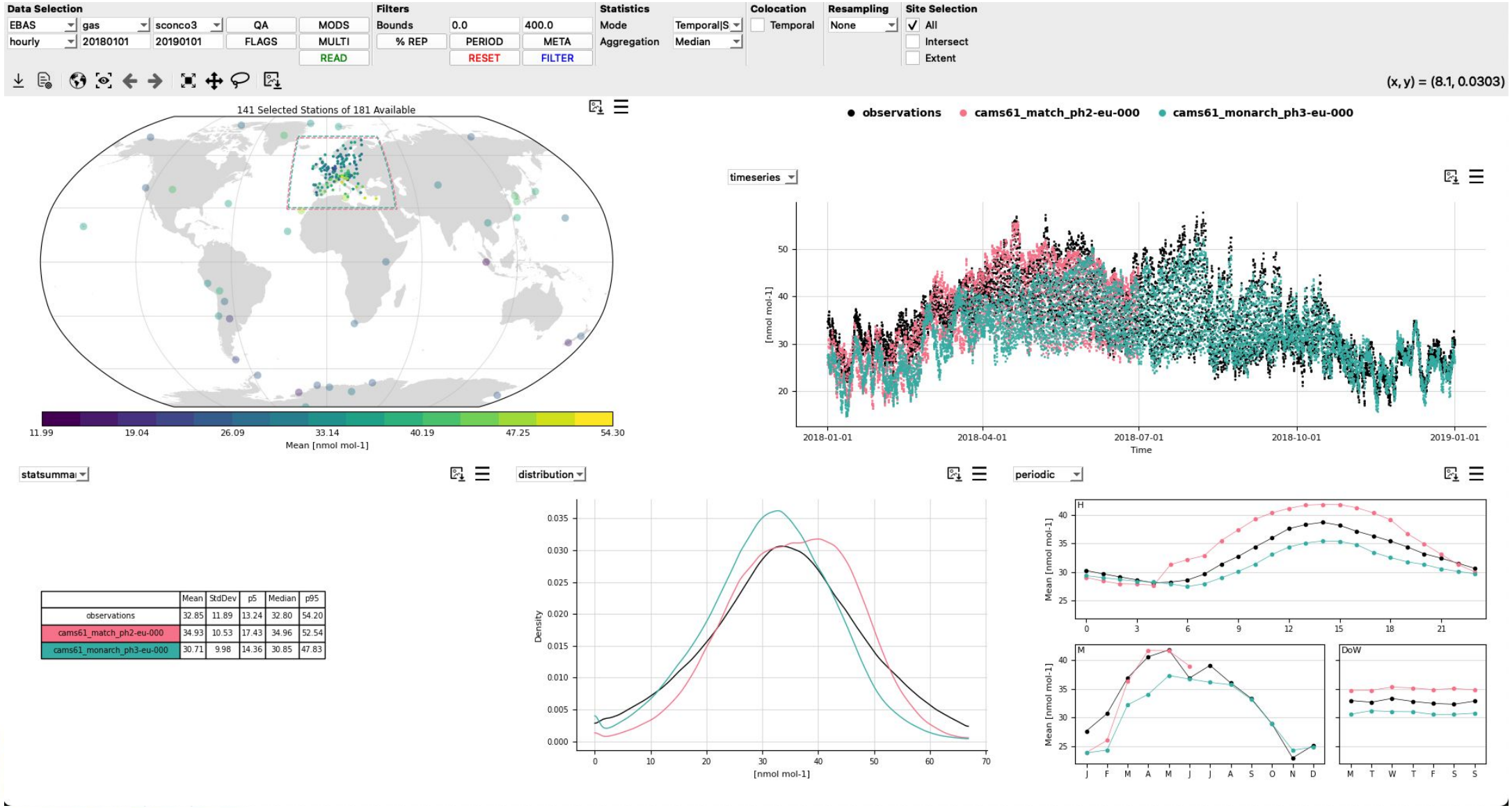
When temporal colocation is active, you will have access to more plot types (scatter, taylor, fairmode-target, and fairmode-statsummary).

Temporal colocation can be set in the configuration file by setting a boolean as follows (default: True):

```
temporal_colocation = False
```

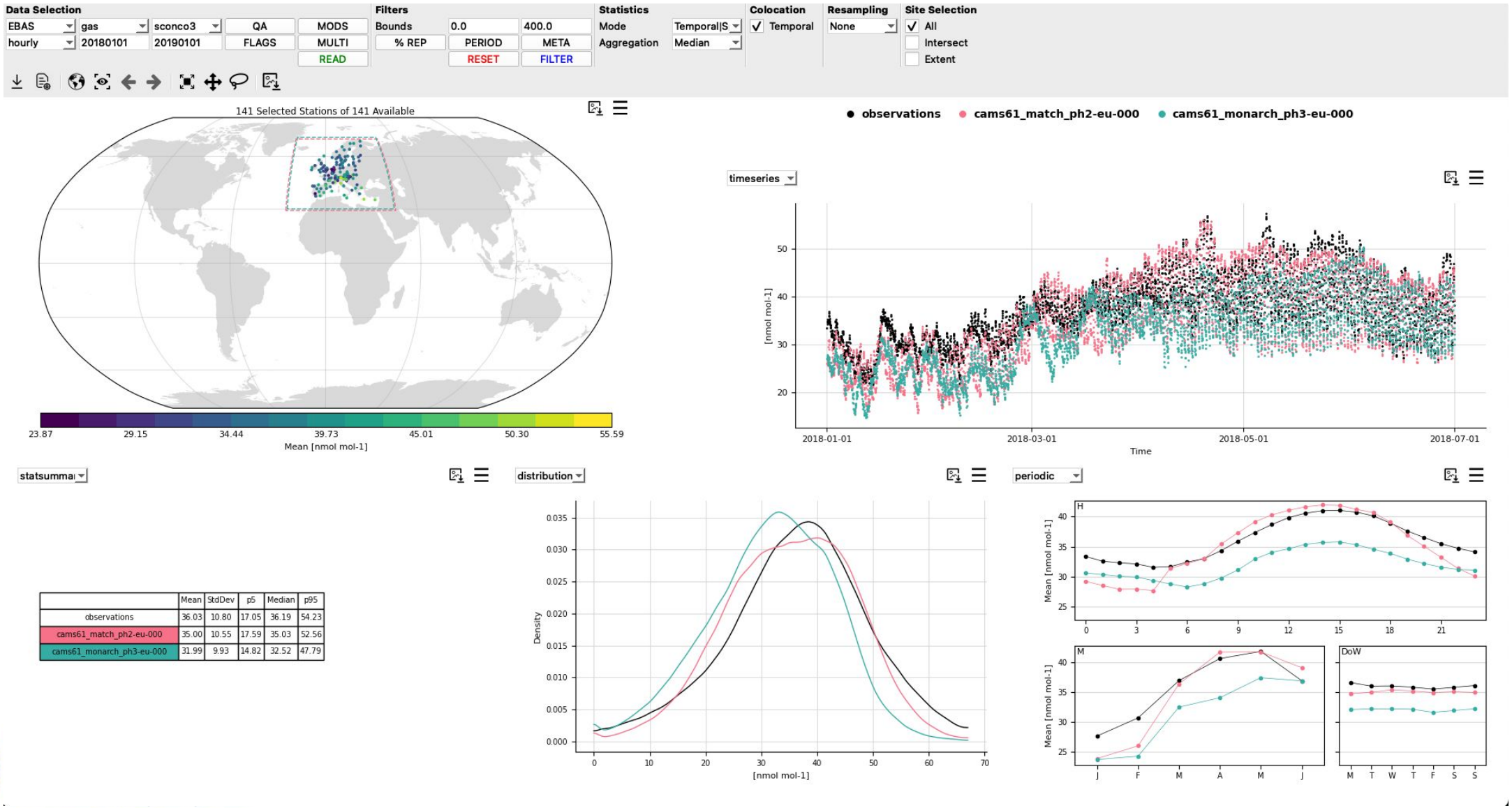
Reference: https://providentia.readthedocs.io/en/latest/Colocation.html

Without temporal colocation



	Mean	StdDev	p5	Median	p95
observations	32.85	11.89	13.24	32.80	54.20
cams61_match_ph2-eu-000	34.93	10.53	17.43	34.96	52.54
cams61_monarch_ph3-eu-000	30.71	9.98	14.36	30.85	47.83

With temporal colocation



	Mean	StdDev	p5	Median	p95
observations	36.03	10.80	17.05	36.19	54.23
cams61_match_ph2-eu-000	35.00	10.55	17.59	35.03	52.56
cams61_monarch_ph3-eu-000	31.99	9.93	14.82	32.52	47.79

Resampling

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Resampling

The resampling section allows the plotted data to be **resampled to a coarser temporal resolution** (by the mean), e.g. hourly to daily, by clicking on a resolution from the drop-down menu. When a resolution is selected, it updates the plots and statistics across the dashboard.

From the configuration file, we need to assign a temporal resolution to the key **resampling_resolution**:

```
resampling_resolution = daily
```

Providentia

Data Selection

EBAS | gas | sconco3 | QA | EXPS
hourly | 20150101 | 20200101 | FLAGS | MULTI
READ

Filters

Bounds | 0.0 | 400.0
% REP | PERIOD | META
RESET | FILTER

Statistics

Mode: Temporal|S
Aggregation: Median

Colocation

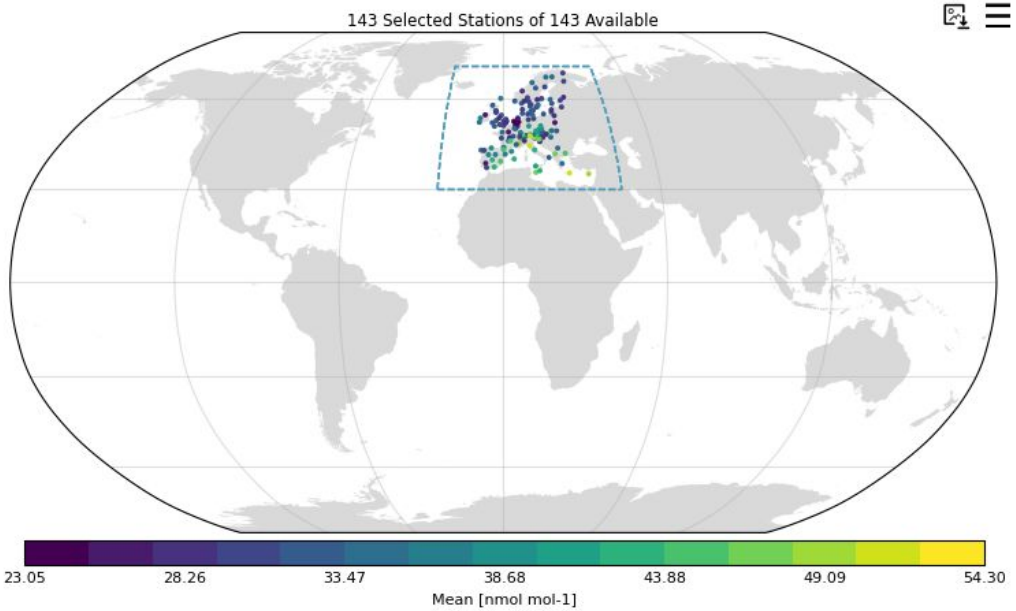
☑ Temporal

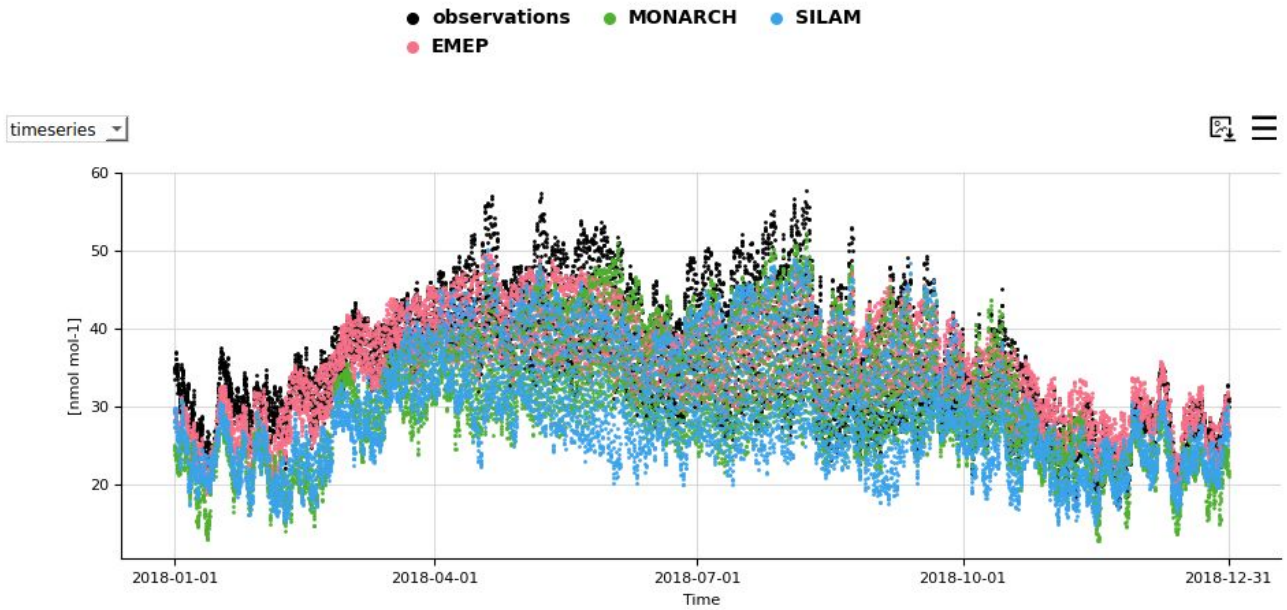
Resampling

None

Site Selection

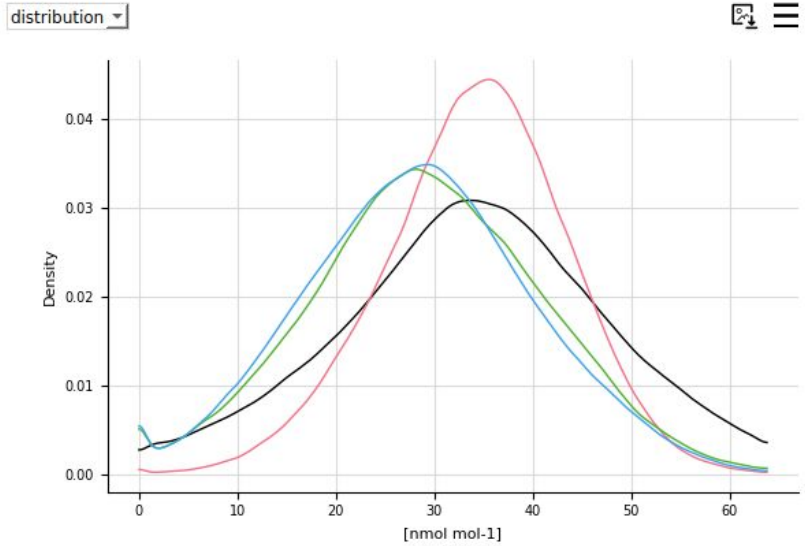
☑ All
☐ Intersect
☐ Extent

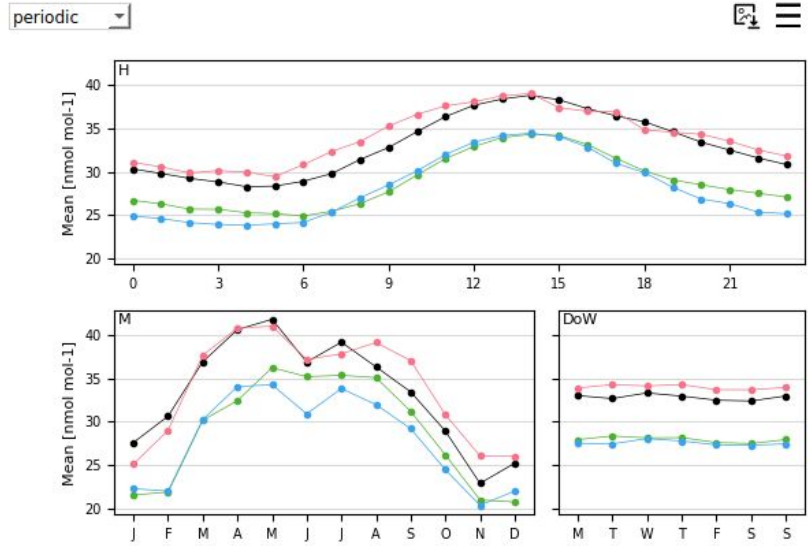




statsummar

	Mean	StdDev	p5	Median	p95
observations	32.91	11.84	13.30	32.80	54.46
EMEP	33.96	8.47	18.77	34.14	47.72
MONARCH	27.88	10.57	12.19	27.44	47.20
SILAM	27.41	10.55	11.02	27.29	47.37





Providentia

Data Selection

EBAS | gas | sconco3 | QA | EXPS
hourly | 20150101 | 20200101 | FLAGS | MULTI
READ

Filters

Bounds | 0.0 | 400.0
% REP | PERIOD | META
RESET | FILTER

Statistics

Mode: Temporal|S
Aggregation: Median

Colocation

☑ Temporal

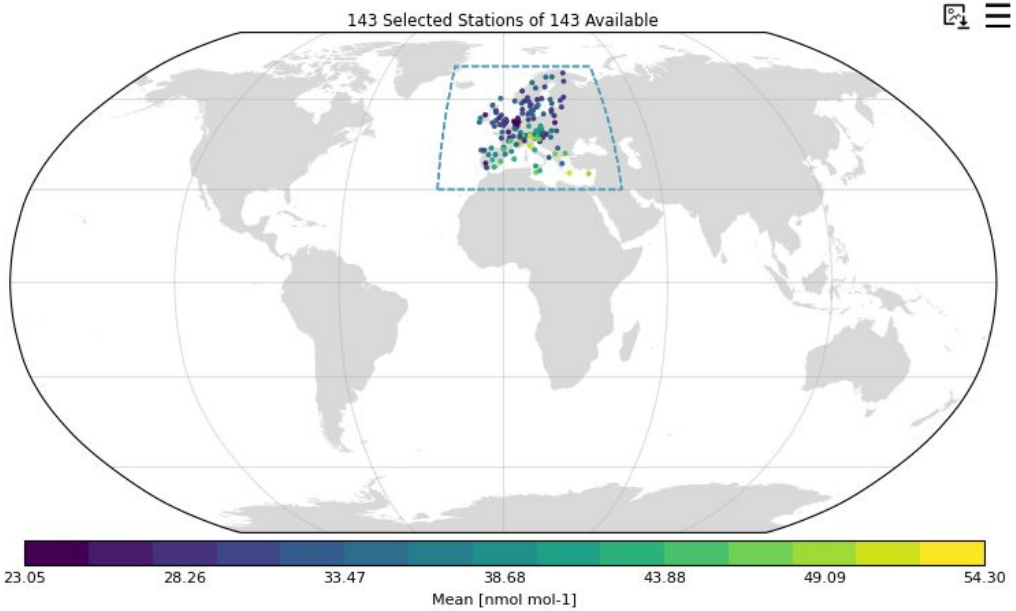
Resampling

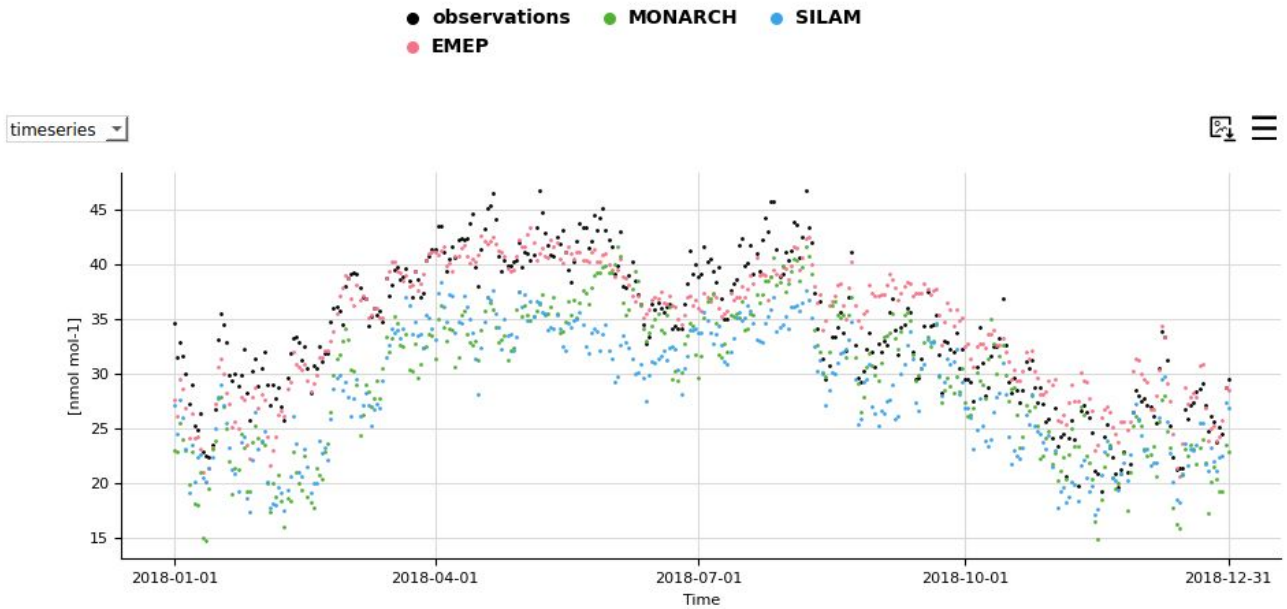
daily

Site Selection

☑ All
☐ Intersect
☐ Extent

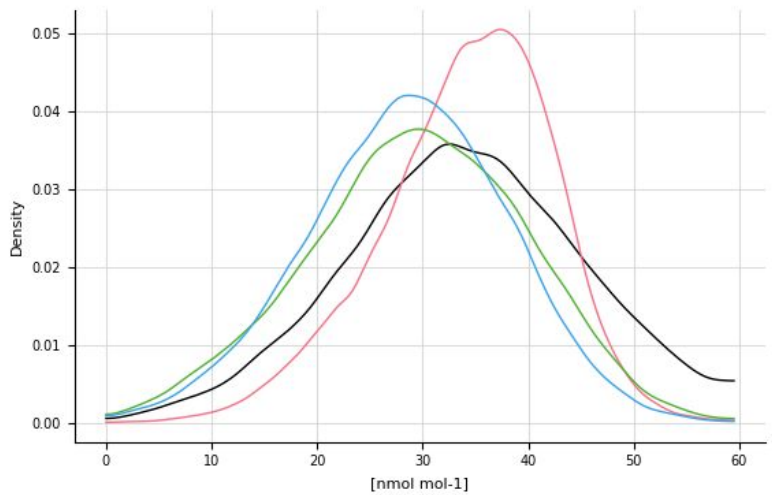
(x, y) = (0.256, 0.92)

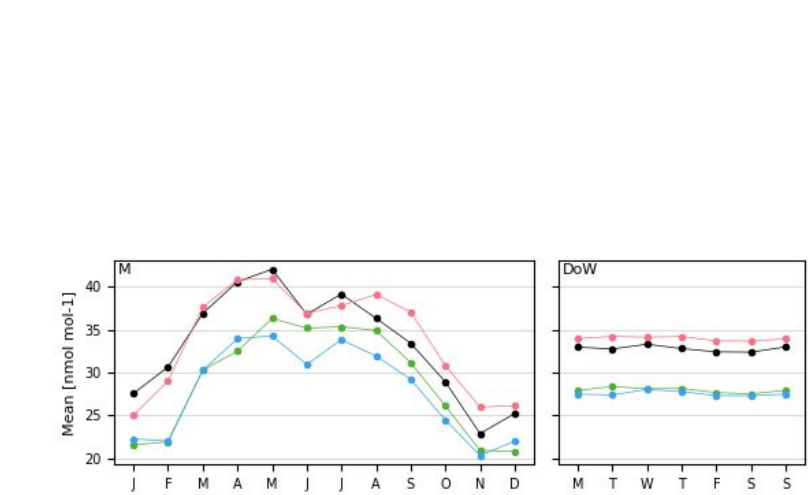




statsummar

	Mean	StdDev	p5	Median	p95
observations	32.91	9.41	17.20	33.11	48.63
EMEP	33.88	7.28	19.98	34.89	44.72
MONARCH	27.86	8.72	13.73	28.71	42.28
SILAM	27.42	8.15	14.17	27.71	40.97





Timeseries chunking

For the timeseries there is an additional selection for the temporal resolution (**chunk resolution**) which works on top of the resampling if active.

For each chunk a statistic can be calculated (**chunk statistic**), allowing to see how that stat evolves over time.

Statistics

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Statistical modes

Most of the plotted data in Providentia depends on certain logical choices made when calculating statistics.

Statistics can be calculated in a variety of ways, and Providentia allows a number of ways to alter the calculation, via 3 statistical modes. These are as follows:

- **Temporal|Spatial (default)**
- **Spatial|Temporal**
- **Flattened**

Statistical modes

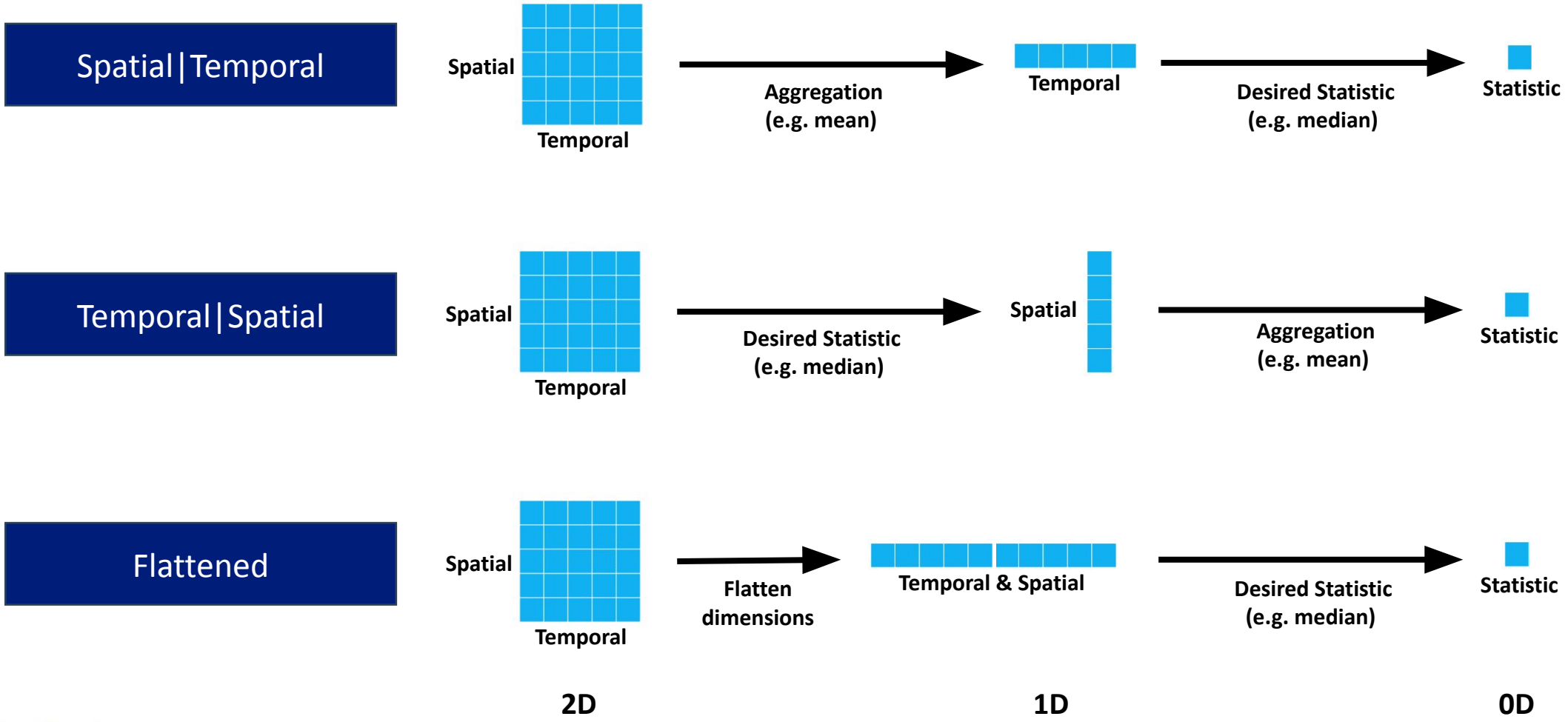
The name of each statistical mode relates to how the dimensions of the selected data are reduced to calculate the statistical metrics, e.g. mean, median etc, going from 2D to 0D.

When selecting data across multiple stations, it has 2 dimensions: **Spatial** and **Temporal**.

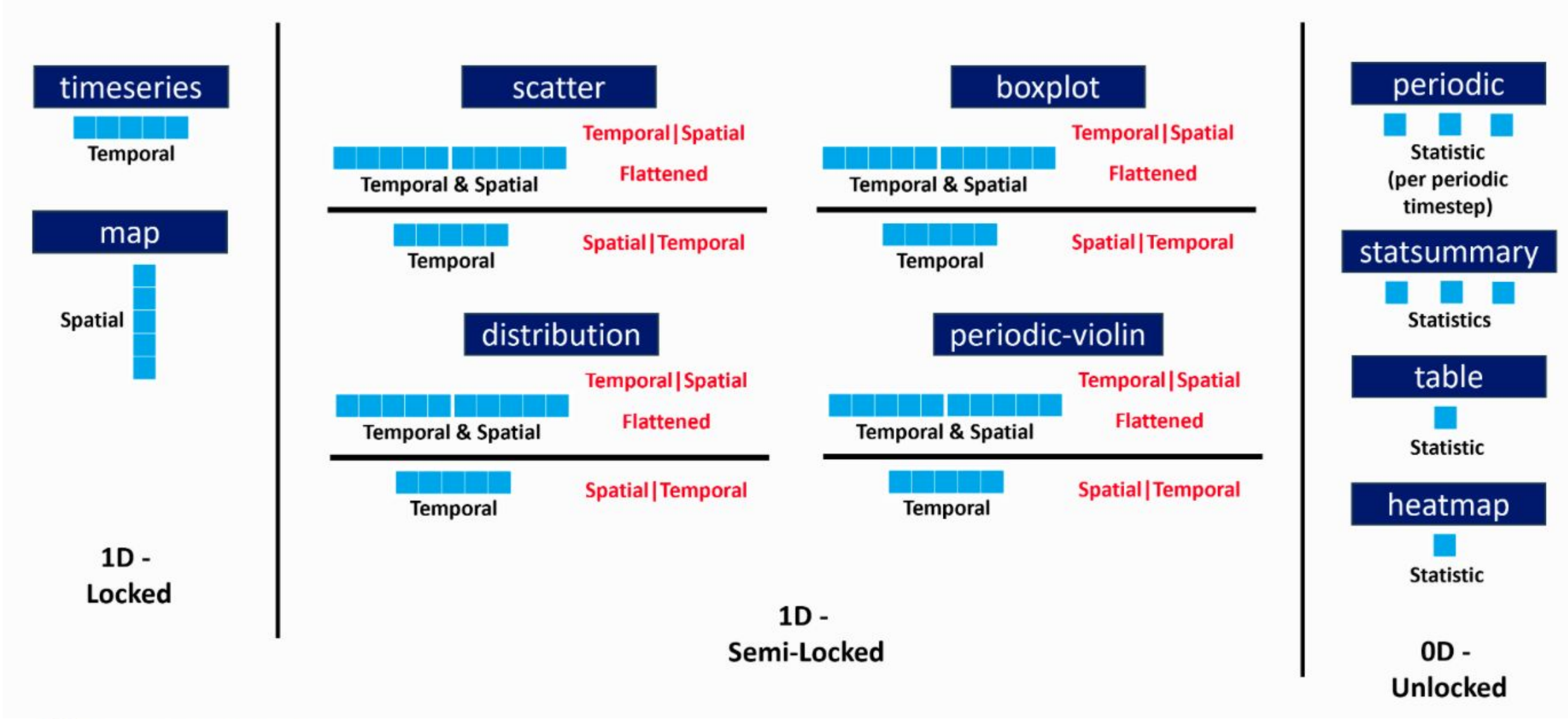
For **Temporal|Spatial**, aggregation is performed first temporally per station (e.g. median), going to 1D, before then calculating the desired statistic across stations (e.g. p50), going to 0D.

For some plot types the full dimensional reduction is not possible, e.g. map. Therefore, they are locked in certain reduction configurations.

Statistical modes



Dimensional reduction per plot type



Statistical modes

On the dashboard, in the statistics menu at the top, the mode and aggregation can be selected via the dropdown menus.

In the configuration files these can be set like so:

statistic_mode = Temporal|Spatial

statistic_aggregation = Median

Note: For the Flattened mode, there is no aggregation statistic.

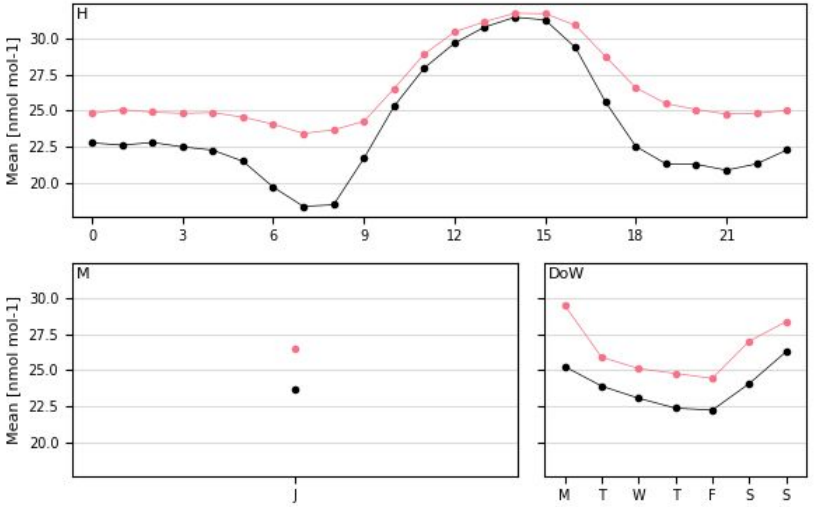
Periodic statistical modes

The periodic plot gives statistical information for grouped data in individual timesteps. Thus it can be seen how each individual timesteps compare for observations vs model/s.

However, statistics can also be calculated which assess the entirety of each periodic cycle, i.e. diurnal, weekly, annual. These statistics are available via the **statsummary**, **table** and **heatmap** plot types.

Note: Only statsummary is available on the dashboard

Periodic plot



Periodic statistical modes

There exist 2 modes for calculating these periodic statistics:

- **Independent (default)**
- **Cycle**

Independent works by calculating the desired statistic (e.g. r) per periodic timestep (i.e. as seen in periodic plot), before aggregating across the timesteps (e.g. median).

Cycle works by aggregating the grouped data per timestep (e.g. mean), before then calculating the desired statistic across the timesteps (e.g. MB).

Periodic statistics

On the dashboard, in the plot options of the **statsummary** plot, the periodic statistic mode and aggregation, and periodic statistics can be selected for specific diurnal, weekly and monthly cycles via the dropdown menus.

In the configuration file these can be set like so:

periodic_statistic_mode = Independent

periodic_statistic_aggregation = Median

For non-dashboard modes, the periodic statistic can be set via adding the specific cycle to the statistic, e.g. **Mean-diurnal,** in the **statsummary** section of **settings/plot_characteristics.yaml**.

Forecast

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Forecast options

Forecast models predict the state of the atmosphere in the near future, most typically on the scale of days. Forecast data comes in a different format to most other kinds of model data, with data per day for the subsequent N days.

This data can be visualised in Providentia in different ways, set by the **forecast** variable in the configuration file with the following valid fields:

- **day, dayN**
- **combined, daily**
- **dailyN**

It is possible to compare different models where one has forecast data and the other does not. In these cases the non-forecast data is treated as day 1 of a forecast for comparison purposes.

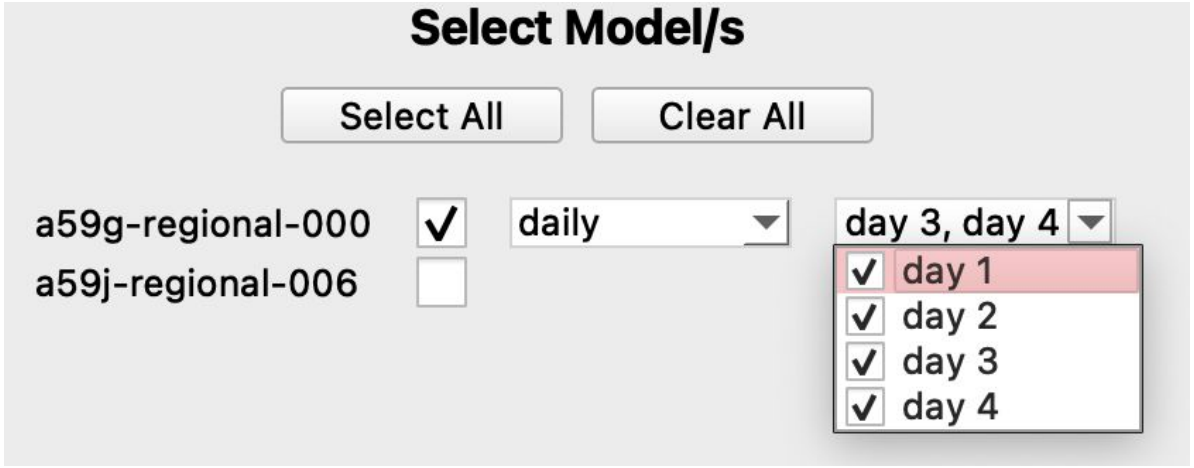
If no forecast option is set for a model with forecast data, data from the first forecast day will simply be taken to represent the model (i.e. treating the data as a non-forecast type).

Forecast data on the dashboard

When using the dashboard, models that have have interpolated forecast data are automatically detected.

When clicking on the **MODS** button on the menu bar, if the model has forecast data, then when you select the model, an extra drop-down menu will appear to the right. This will allow you to select the forecast option, i.e. combined, daily or day .If no option is set, data from the first forecast day will be taken to represent the model (i.e. treating the data as a non-forecast type).

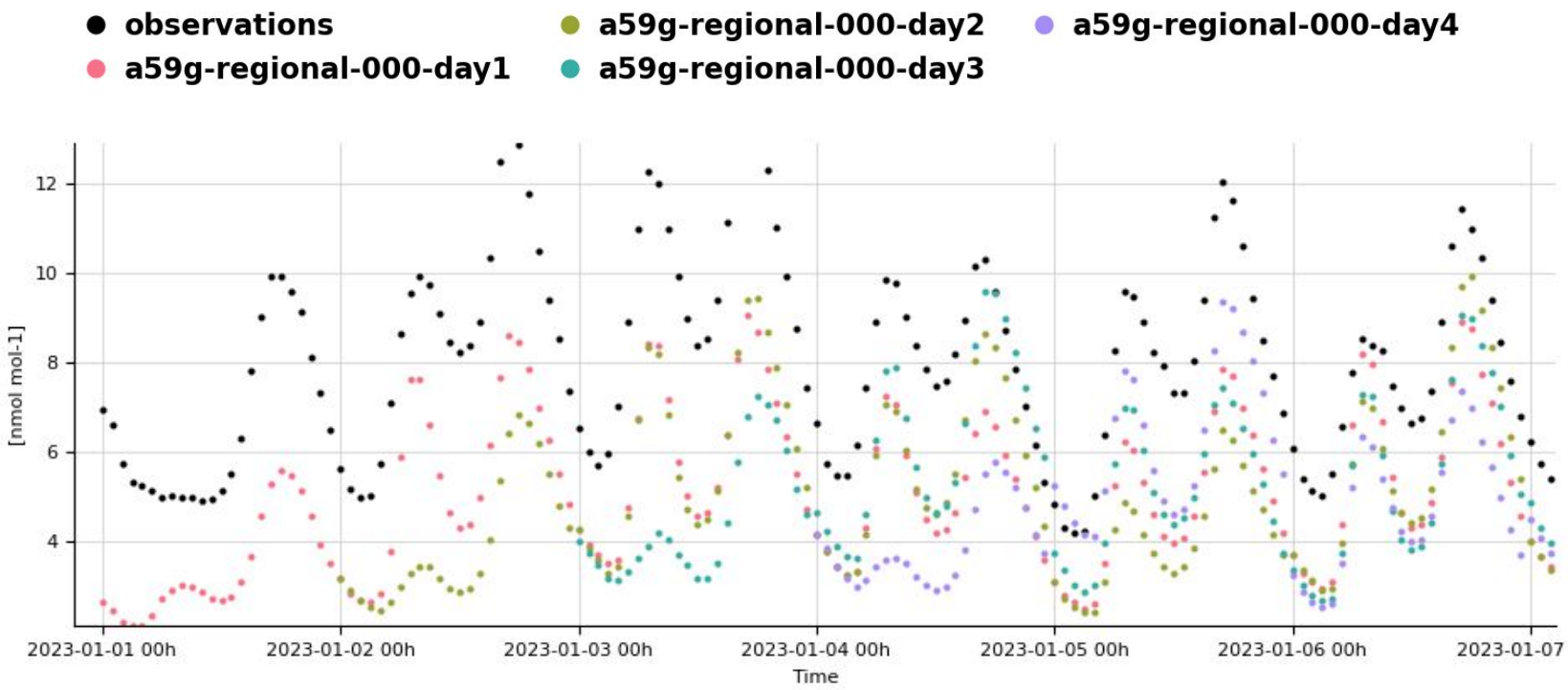
If a forecast option is selected, another drop-down menu will then appear to the right of that, allowing you to select specific forecast days. If no specific forecast days are selected but a forecast option is, then all forecast days will be utilised.



Interactive selection of model forecast data

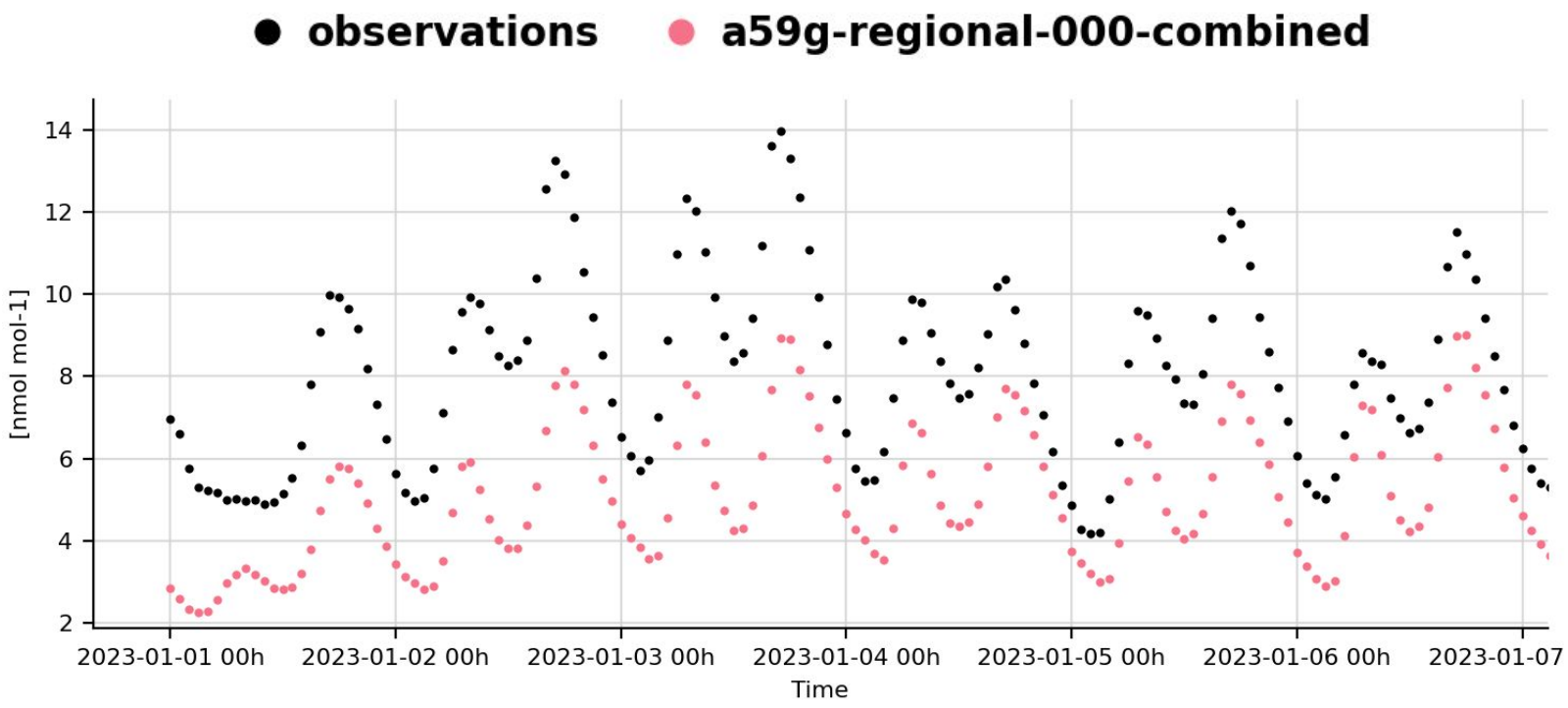
Day forecast

The **day** option is used when wanting to compare different forecast days. The effect this has in Providentia is essentially to treat each forecast day as an independent model. Thus statistics for each forecast day can be directly compared.



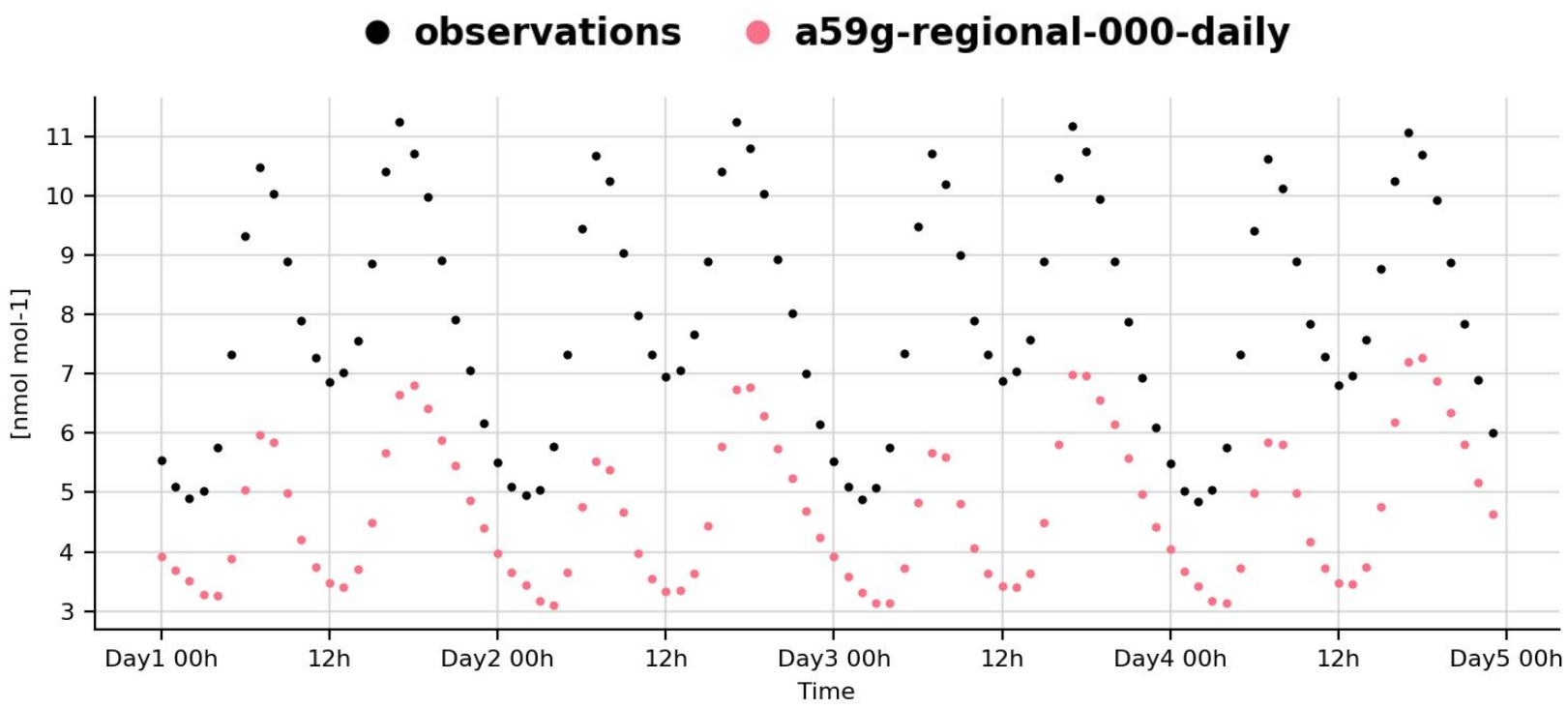
Combined forecast

The **combined** option is used when wanting to integrate data across all forecast days, providing statistics which represent the average state across all forecast days. The timeseries plot shows the average timeseries across all forecast days, for the entire time range.



Daily forecast

The **daily** option is also used when wanting to integrate data across all forecast days, however the timeseries plot is transformed to display the averaged data per forecast day, i.e. the plotted timeseries data for forecast day 1 in this case would be the average day 1 forecast across the entire time range.



Dashboard practical

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Dashboard practical

1. Perform a visualisation for **hourly ozone** from **EEA,** in **January 2023**, using **GHOST version 1.6**, for the **a59g-regional-000** model, for **Spanish stations** focused on the **bias** between observations and model, making a **map r bias**, **timeseries bias, periodic r bias**, **distribution bias,** and **statsummary bias**.

2. Perform a visualisation for **hourly ozone** from **EEA,** in **January 2023**, using **GHOST version 1.6,** for the **a59j-regional-006 and a59g-regional-000** models, which keeps only stations with **>= 40% total representativity**, for strictly **daytime** periods, at stations **1km from the coast**.

3. Perform a **forecast day** visualisation for **hourly nitrogen dioxide** from **EEA,** in **January 2023**, using **GHOST version 1.6,** for the **a59g-regional-000** model**,** only keeping stations along the **coast of the United Kingdom and Ireland**, increase the selected map point size, **resample** the data to 3 hourly, and **save the data** to a netCDF file.

BREAK
10 minutes

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Report

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Choosing the plots

Users need to define the **report_type** in the configuration file, which will be linked to the plot types that appear in **settings/report_plots.yaml.**

configurations/configuration_name.conf

```
[PRV_sconco3_CHIMERE]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180601
model = cams61_chimere_ph2-eu-000 (CHIMERE)
report_type = docs
report_summary = True
report_stations = True
report_title = Providentia Report
report_filename = CHIMERE
```

settings/report_plots.yaml

```
"docs": ["map-Mean", "timeseries", "periodic-Mean",
"periodic-violin", "boxplot", "distribution",  "scatter",
"heatmap-RMSE", "table-RMSE", "statsummary",
"taylor-r",  "fairmode-target",
"fairmode-statsummary", "contingencytable",
"metadata"]
```

Summary or station plots?

We can show data for all stations (report_summary=True) or per station (report_station=True). It is also possible to show certain plots as summary and others only per station by using a dictionary to define the report type in **settings/report_plots.yaml.**

configurations/configuration_name.conf

```
[PRV_sconco3_CHIMERE]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180601
model = cams61_chimere_ph2-eu-000 (CHIMERE)
report_type = docs
report_summary = True
report_stations = True
report_title = Providentia Report
report_filename = CHIMERE
```

settings/report_plots.yaml

```
"docs": ["map-Mean", "timeseries", "periodic-Mean",
"periodic-violin", "boxplot", "distribution",  "scatter",
"heatmap-RMSE", "table-RMSE", "statsummary", "taylor-r",
"fairmode-target", "fairmode-statsummary", "contingencytable",
"metadata"]
```

```
"docs":
{"summary": ["map-Mean", "timeseries", "periodic-Mean",
                "periodic-violin", "boxplot", "distribution",
                "scatter", "heatmap-RMSE", "table-RMSE",
                "statsummary", "taylor-r", "fairmode-target",
                "fairmode-statsummary", "contingencytable",
                "metadata"],
"station": ["contingencytable"]
}
```

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Report specific configuration options

Parameter	Description	Default
report_type	Type of report to generate that defines which plots the report will contain, from the options given in report_plots.yaml.	standard
report_summary	Boolean variable to set if you wish to make specific plots for each station in subsection.	True
report_stations	Boolean variable to set if you wish to make summary plots across station subsection.	False
report_title	The header in the first page of the report (as in the PDF).	Providentia Report
report_filename	The filename of the report or the path to create the report (as in the PDF).	PROVIDENTIA_Report
harmonise_stations	Boolean variable to set if you wish to harmonise axes limits across stations for stations report.	True
harmonise_summary	Boolean variable to set if you wish to harmonise axes limits across subsections for summary report.	True

Plot types

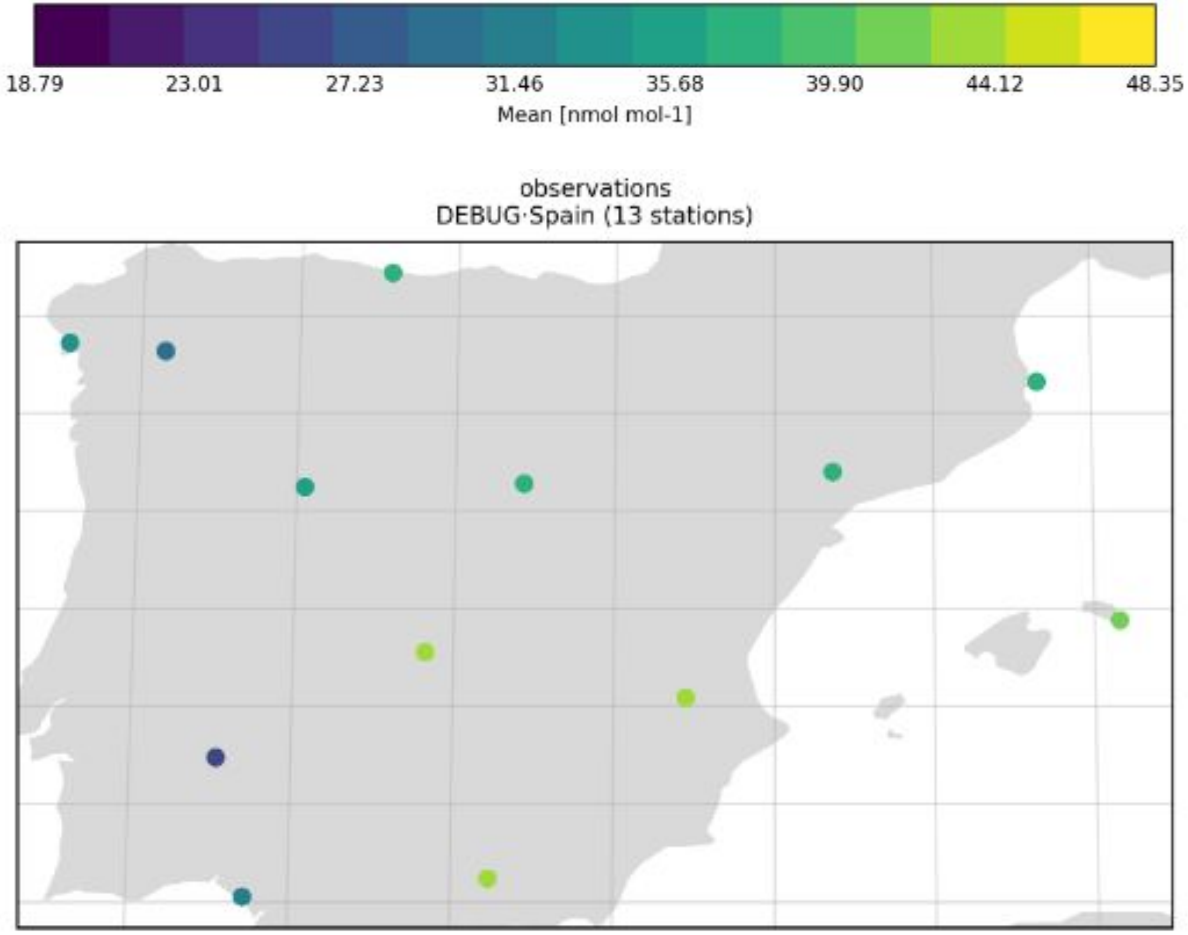
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Plot types

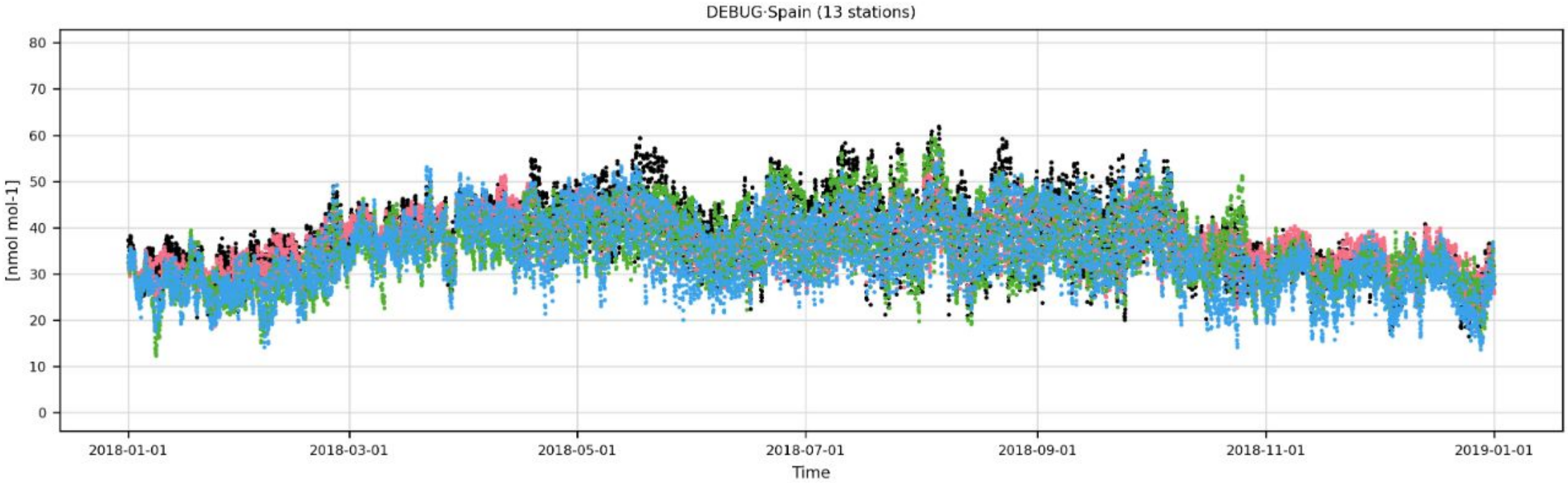
- Map (map)
- Metadata summary (metadata)
- Timeseries (timeseries)
- Periodic plot (periodic)
- Periodic violin plot (periodic-violin)
- Box plot (boxplot)
- Distribution plot (distribution)
- Scatter plot (scatter)
- Heat map (heatmap) - Not available in dashboard
- Table that gives one statistic per subsection per model (table) - Not available in dashboard
- Table that gives multiple statistics per model (statsummary)
- Taylor Diagram (taylor)
- FAIRMODE target plot (fairmode-target)
- FAIRMODE statistics summary plot (fairmode-statsummary)
- Contingency table (contingencytable)

Reference: https://providentia.readthedocs.io/en/latest/Plot-types-and-options.html

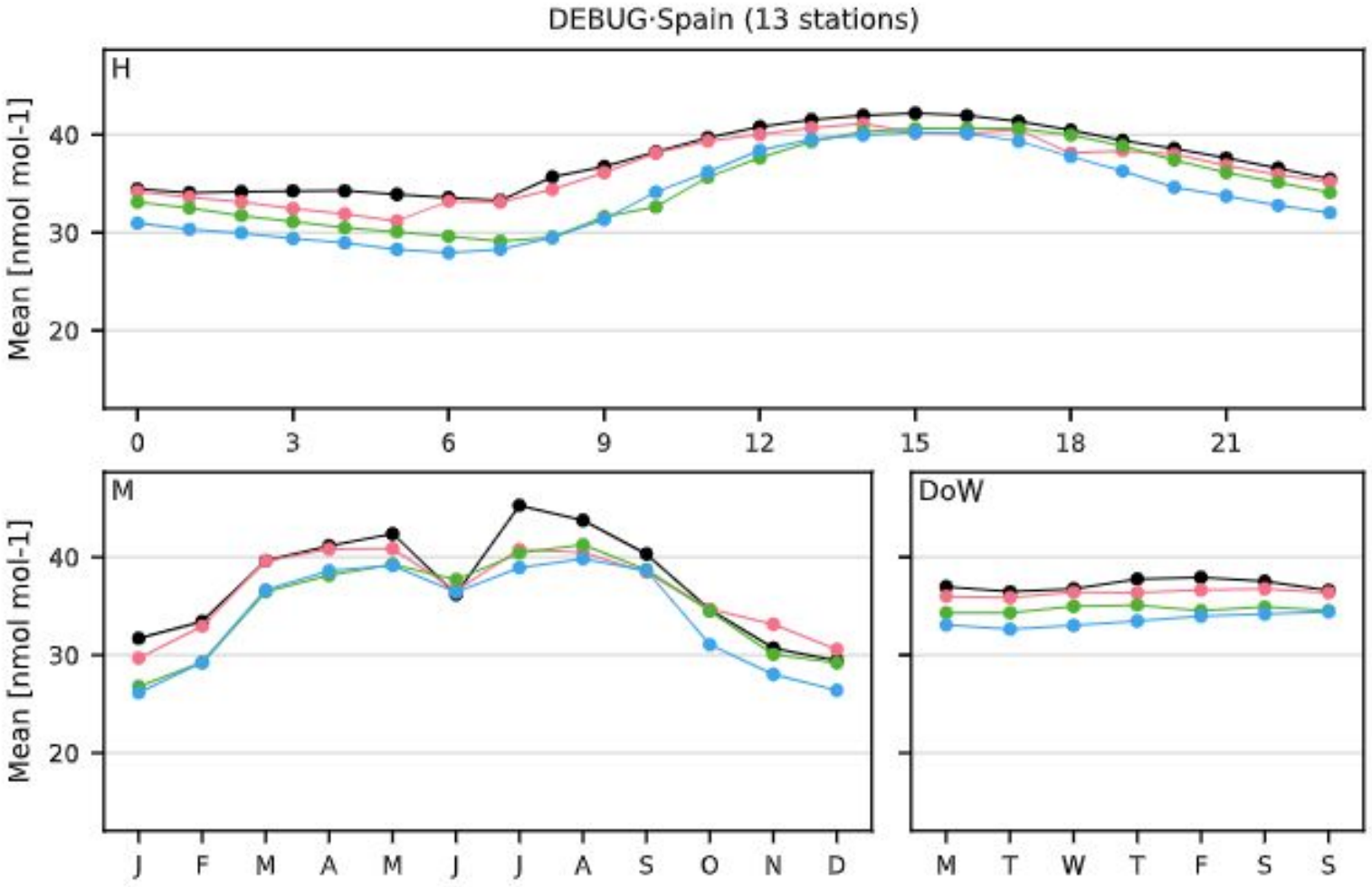
map-[stat]


18.79
23.01
27.23
31.46
35.68
39.90
44.12
48.35
Mean [nmol mol-1]
observations
DEBUG·Spain (13 stations)

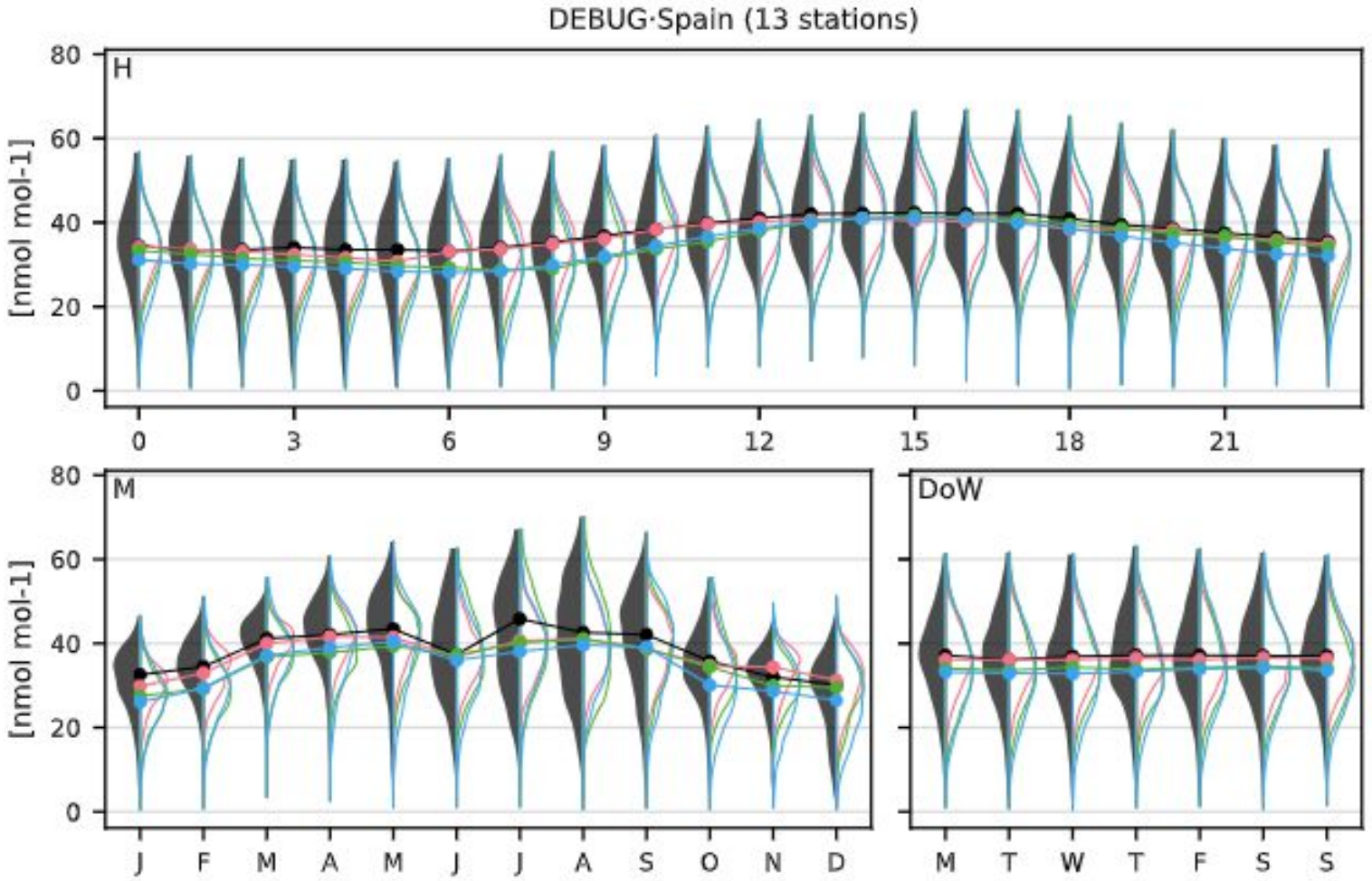
timeseries / timeseries-[stat]-[resolution]



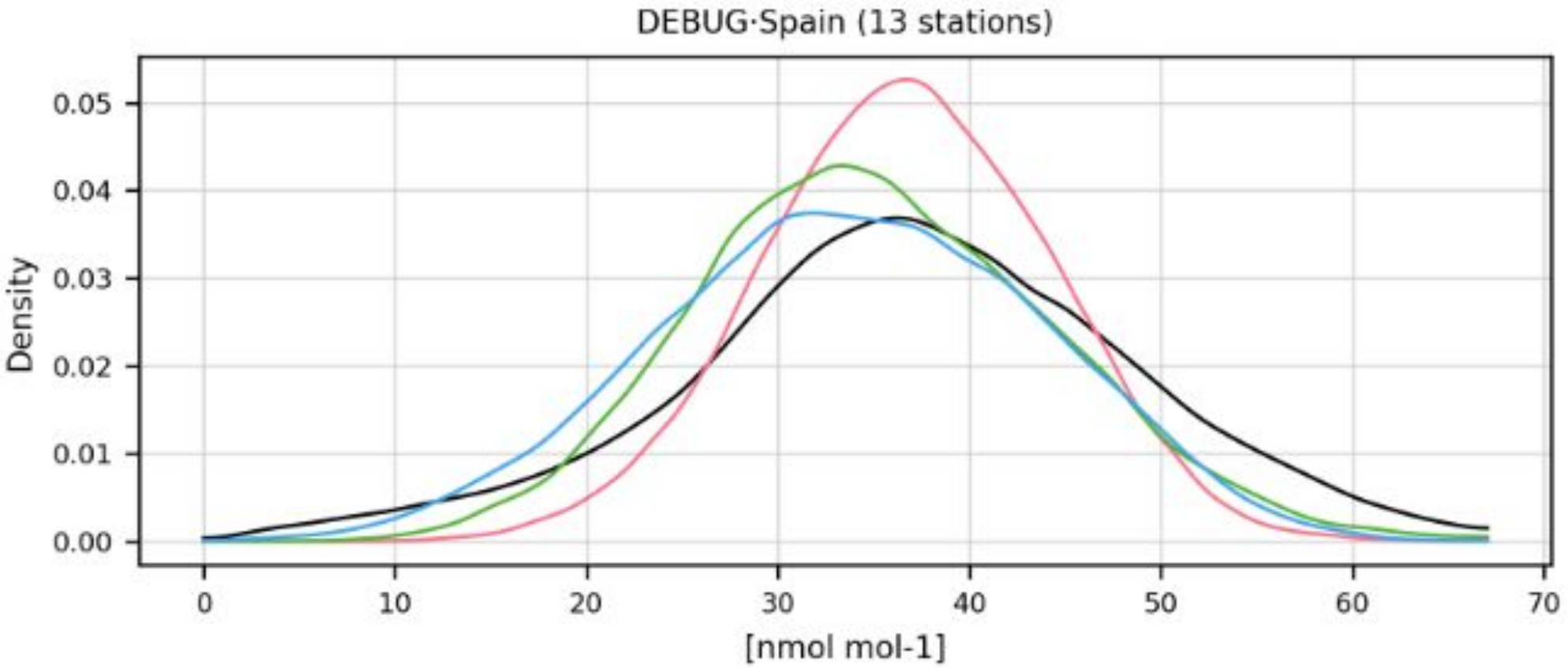
periodic-[stat]



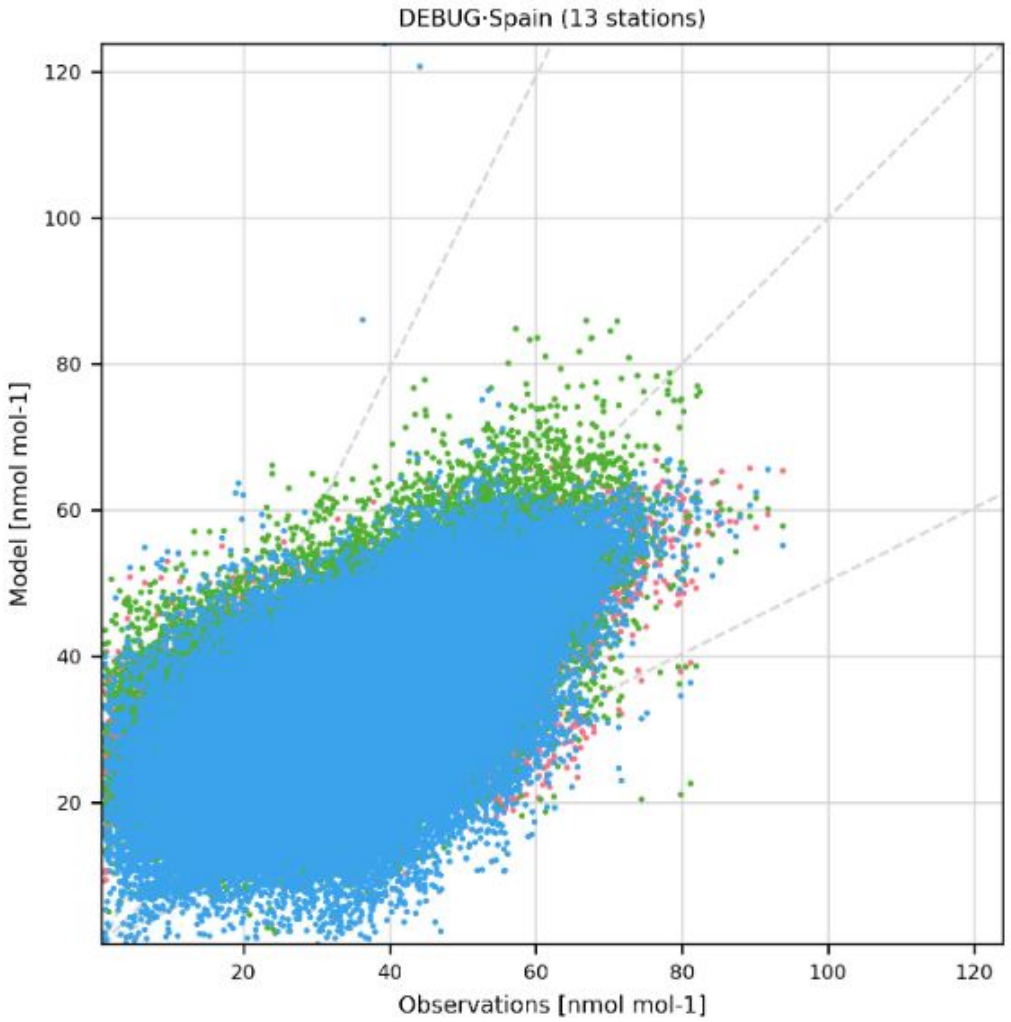
periodic-violin



distribution



scatter


DEBUG·Spain (13 stations)
Model [nmol mol-1]
Observations [nmol mol-1]
20
40
60
80
100
120

heatmap-[stat]

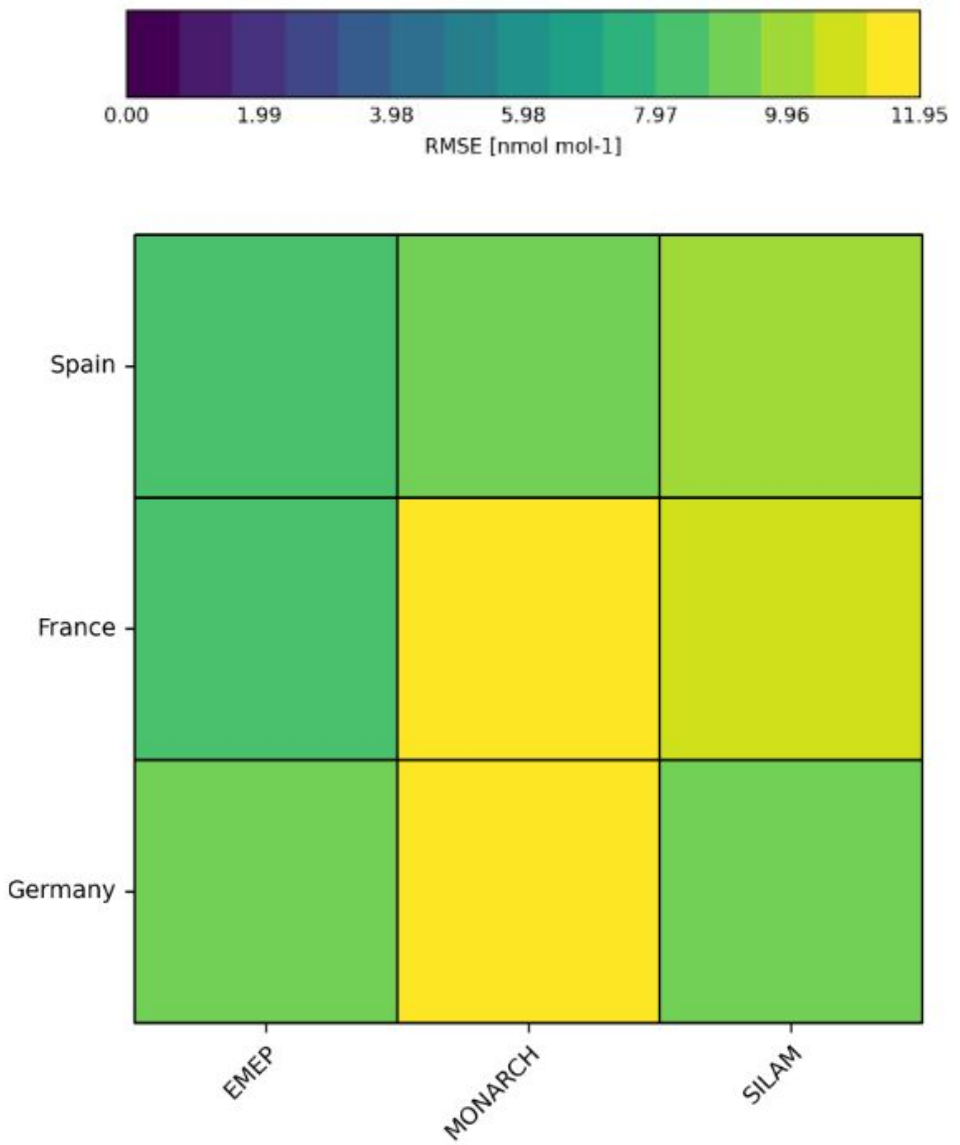
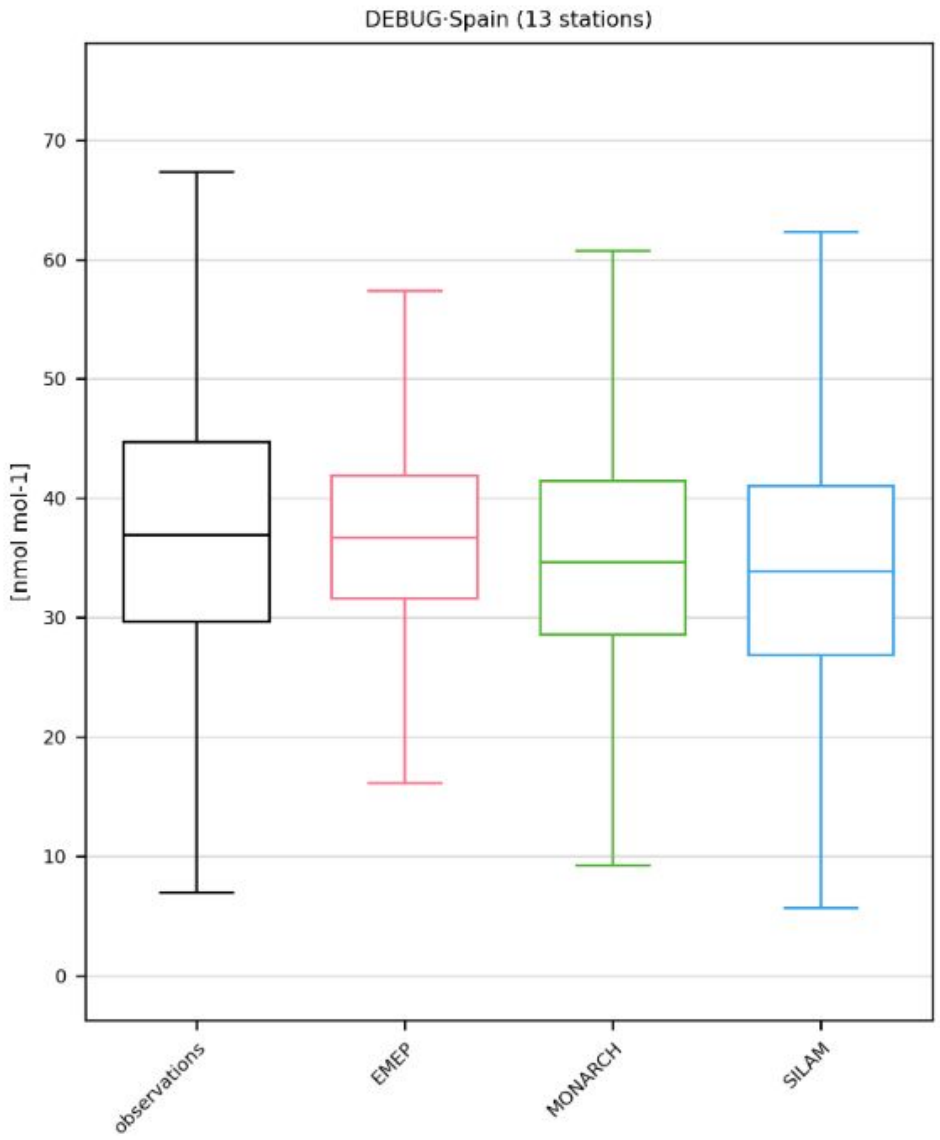


table-[stat]

	EMEP	MONARCH	SILAM
Spain	8.32	9.28	9.77
France	8.55	11.20	10.82
Germany	9.46	11.95	9.25

boxplot


DEBUG·Spain (13 stations)
[nmol mol-1]
0
10
20
30
40
50
60
70
observations
EMEP
MONARCH
SILAM

statsummary

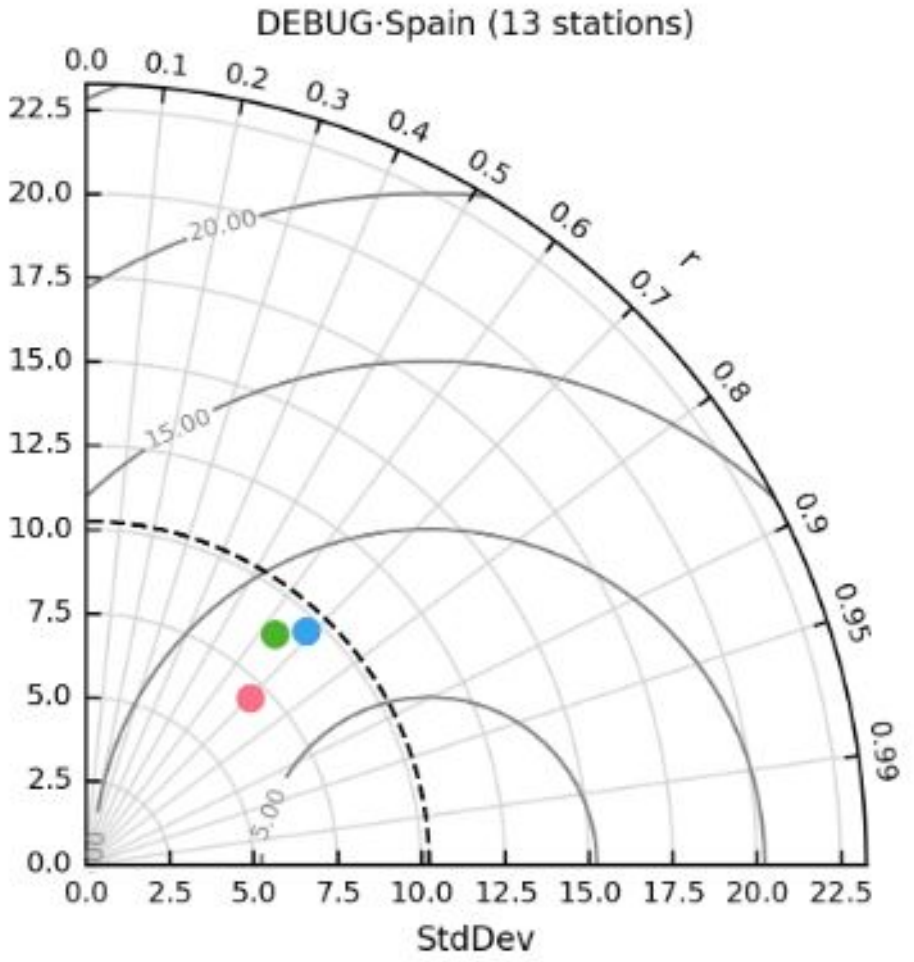
		Mean	StdDev	p5	Median	p95
Spain	observations	37.14	10.24	22.13	37.14	55.03
	EMEP	36.30	6.98	24.61	36.43	48.10
	MONARCH	34.56	8.88	21.43	33.95	51.45
	SILAM	33.64	9.58	18.37	33.25	49.69
France	observations	34.26	11.73	13.65	34.00	57.00
	EMEP	35.80	7.98	20.89	35.86	47.50
	MONARCH	29.35	10.81	11.48	28.21	46.11
	SILAM	29.33	10.80	11.37	28.76	47.61
Germany	observations	33.67	14.89	11.20	34.90	58.30
	EMEP	34.52	10.08	17.48	35.58	48.91
	MONARCH	26.74	12.56	6.73	27.10	47.03
	SILAM	28.81	11.95	8.92	29.04	48.23

metadata

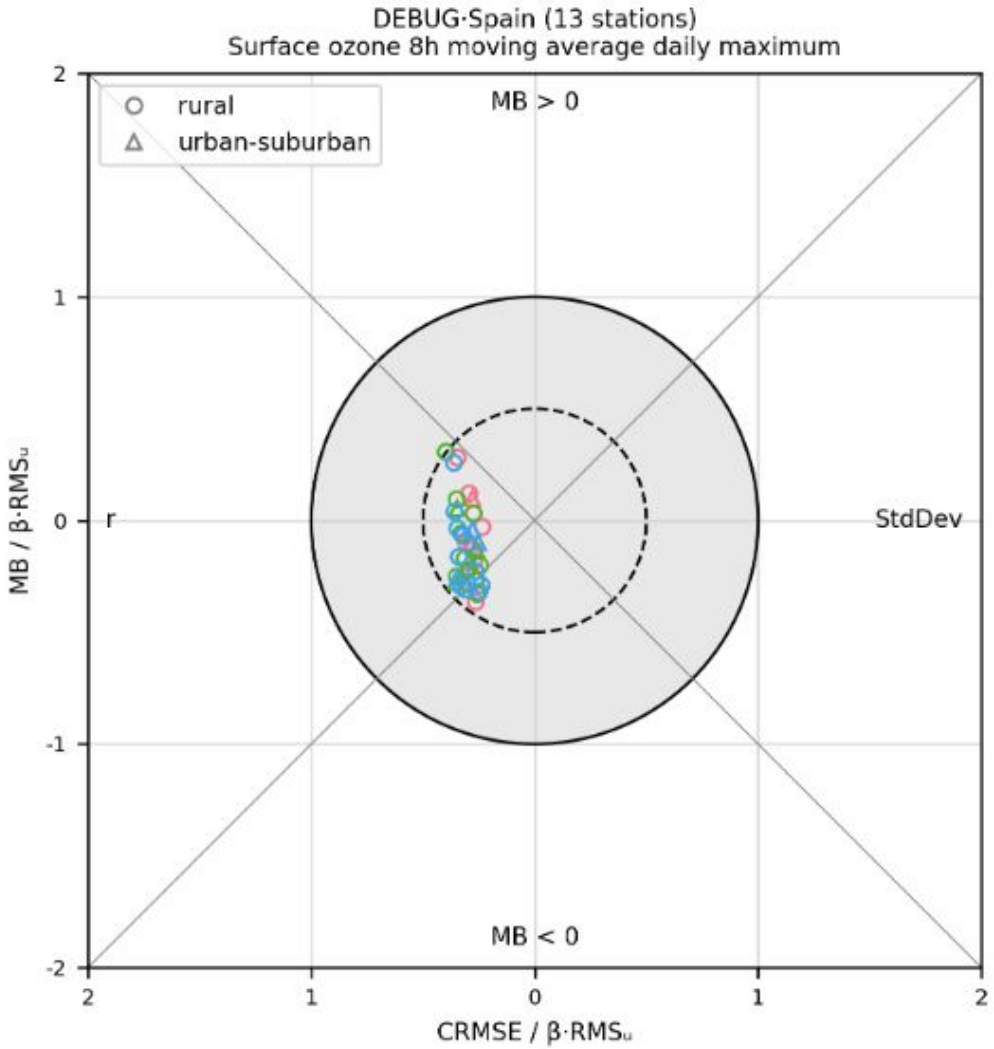
DEBUG·Spain (13 stations)

13 Stations
Median Measurement Altitude: 506.00 m
Median To Coast: -55.00 km
Median Population Density: 0.0 xx/km-2
Country: Spain
Area Classification: rural:76.9%,
urban-suburban:23.1%
Station Classification: background:76.9%,
nan:23.1%
MODIS Land Use: 4 uniques
GHSL Classification: low density rural:23.1%,
very low density rural:76.9%
Method: ultraviolet photometry
Instrument: nan

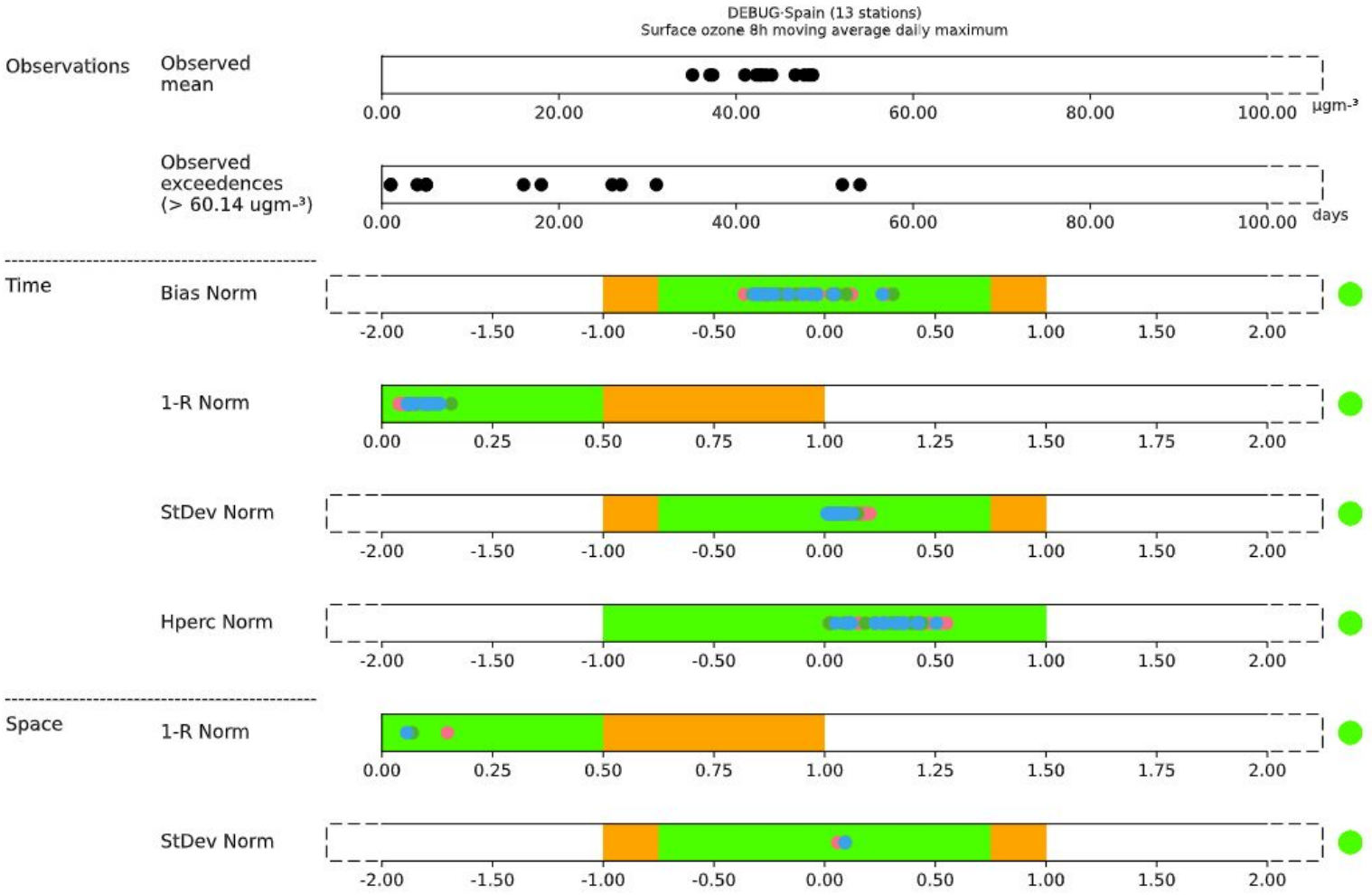
taylor-[stat]


DEBUG·Spain (13 stations)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.95
0.99
r
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
20.00
15.00
5.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
StdDev

fairmode-target



fairmode-statsummary



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

contingencytable

ES0001R_UVP, San Pablo De Los Montes (-4.35, 39.55)

	EMEP					
observations	Good	Fair	Moderate	Poor	Very poor	Extremely poor
Good	32	249	0	0	0	0
Fair	208	5934	24	0	0	0
Moderate	6	1858	278	0	0	0
Poor	0	22	34	0	0	0
Very poor	0	0	0	0	0	0
Extremely poor	0	0	0	0	0	0

Statistics in plots

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

-[stat]

It must be added to create **maps, periodic plots, heatmaps, tables and Taylor diagrams**.

Basic statistics

Statistic	Meaning
Mean	Mean
StdDev	Standard deviation
Var	Variance
Min	Minimum
Max	Maximum
Data%	Data availability
Exceedances	Number of exceedances
p1, p5, p10, p25, p50, p75, p90, p95, p99	Percentiles
NStations	Number of stations
MDA8	Daily maximum 8 hour average

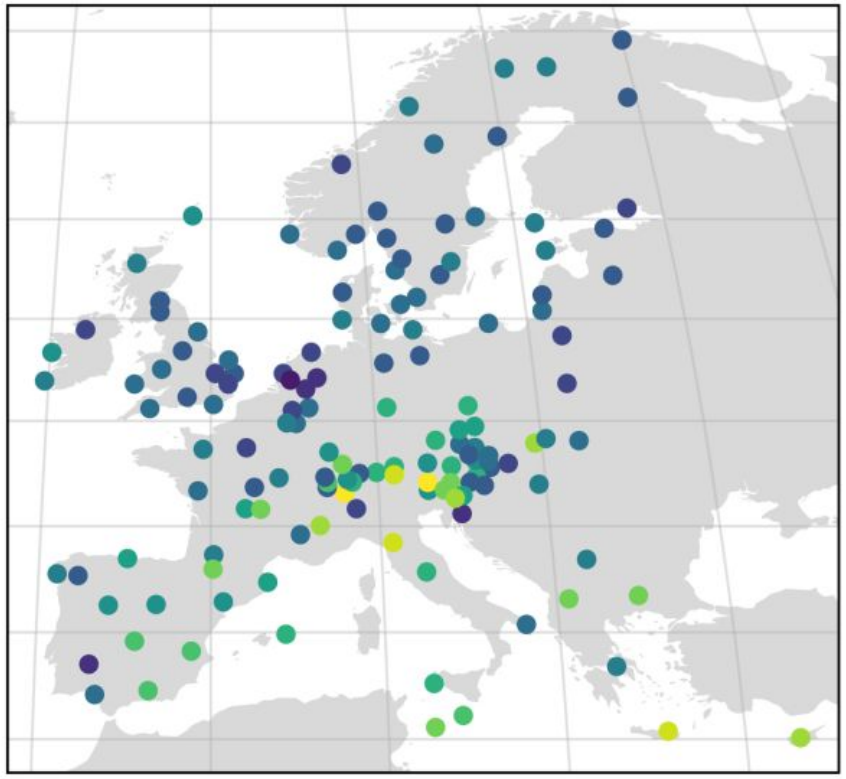
Definitions here

Model bias statistics

Statistic	Meaning
MB	Mean bias
NMB	Normalised mean bias
ME	Mean error
NME	Normalised mean error
MNB	Mean normalised bias
MNE	Mean normalised error
MFB	Mean fractional bias
MFE	Mean fractional error
RMSE	Root mean square error
NRMSE	Normalised root mean square error
COE	Coefficient of efficiency
FAC2	Fraction of model values within a factor of two of observed values
IOA	Index of agreement
R	Pearson correlation coefficient
R^2	Coefficient of determination
UPA	Unpaired peak accuracy

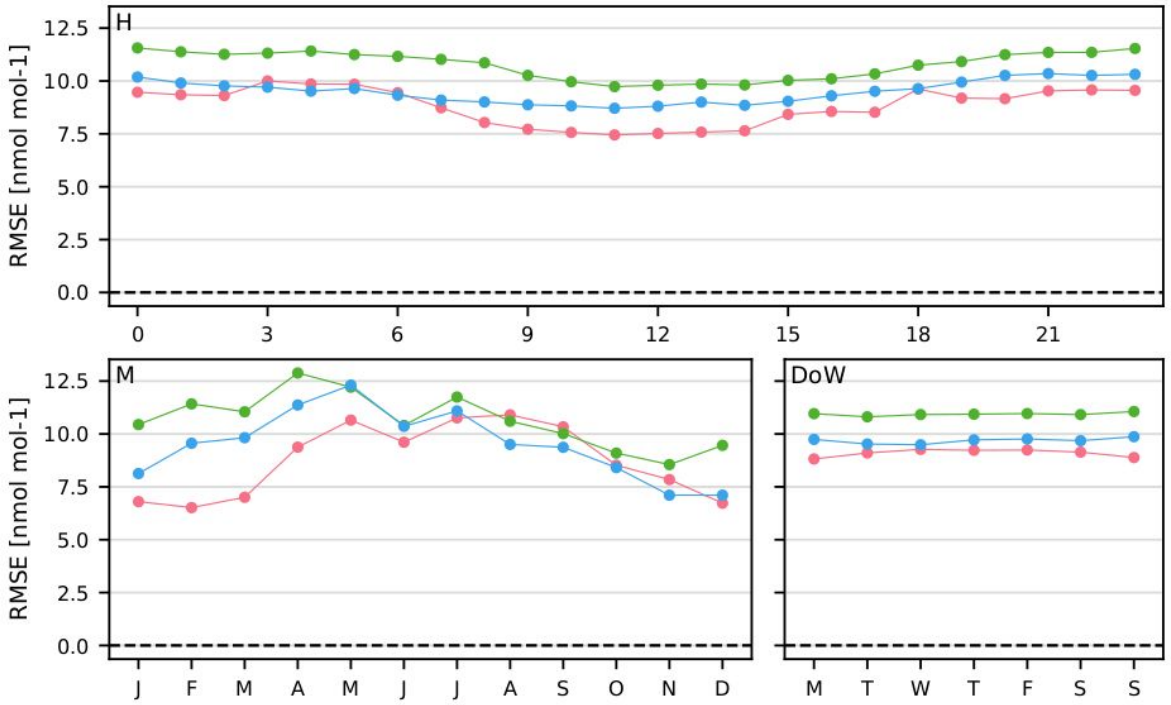
-[stat]

map-Mean

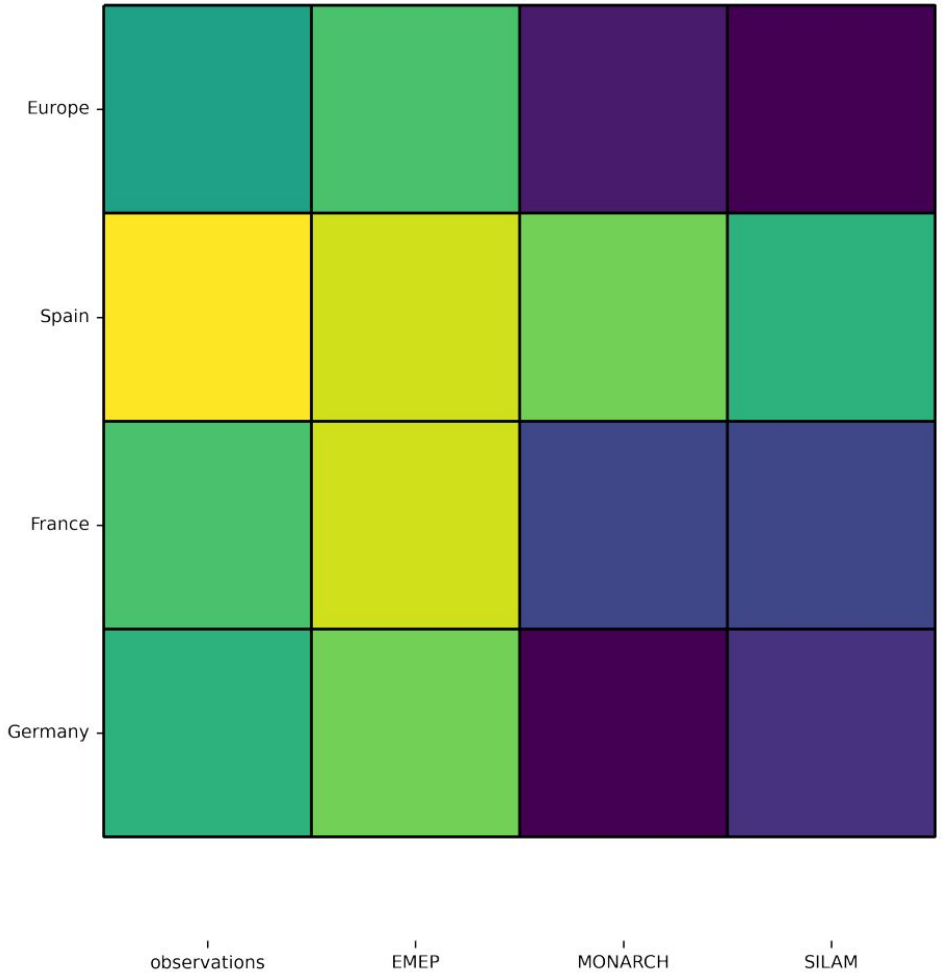


-[stat]

periodic-RMSE



heatmap-Mean



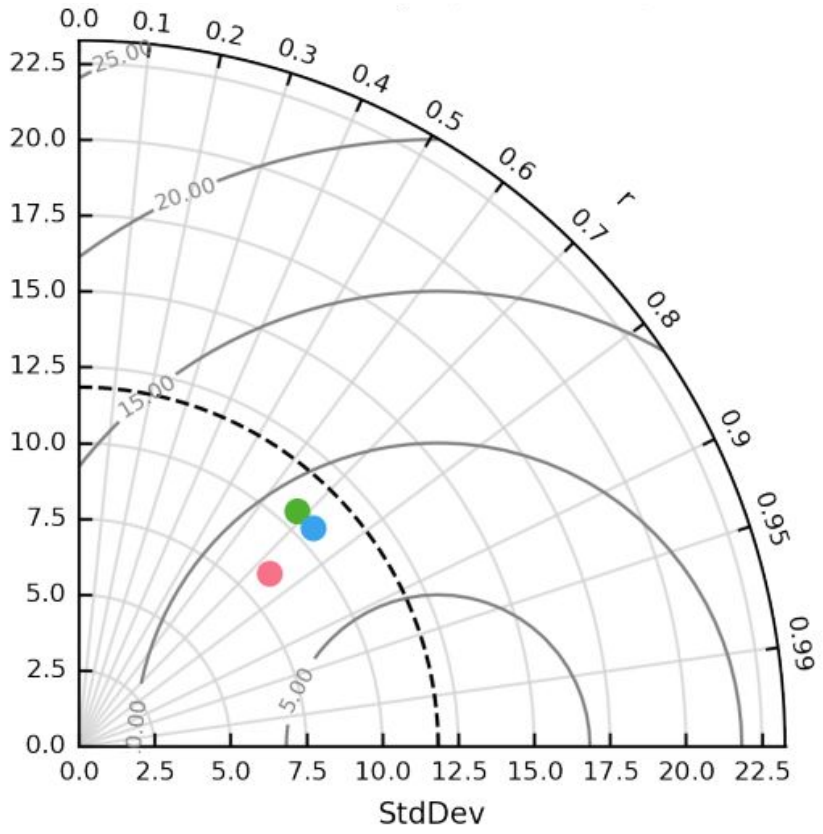
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

-[stat]

table-Data%

	observations	EMEP	MONARCH	SILAM
Europe	19.16	19.16	19.16	19.16
Spain	19.56	19.56	19.56	19.56
France	19.29	19.29	19.29	19.29
Germany	17.41	17.41	17.41	17.41

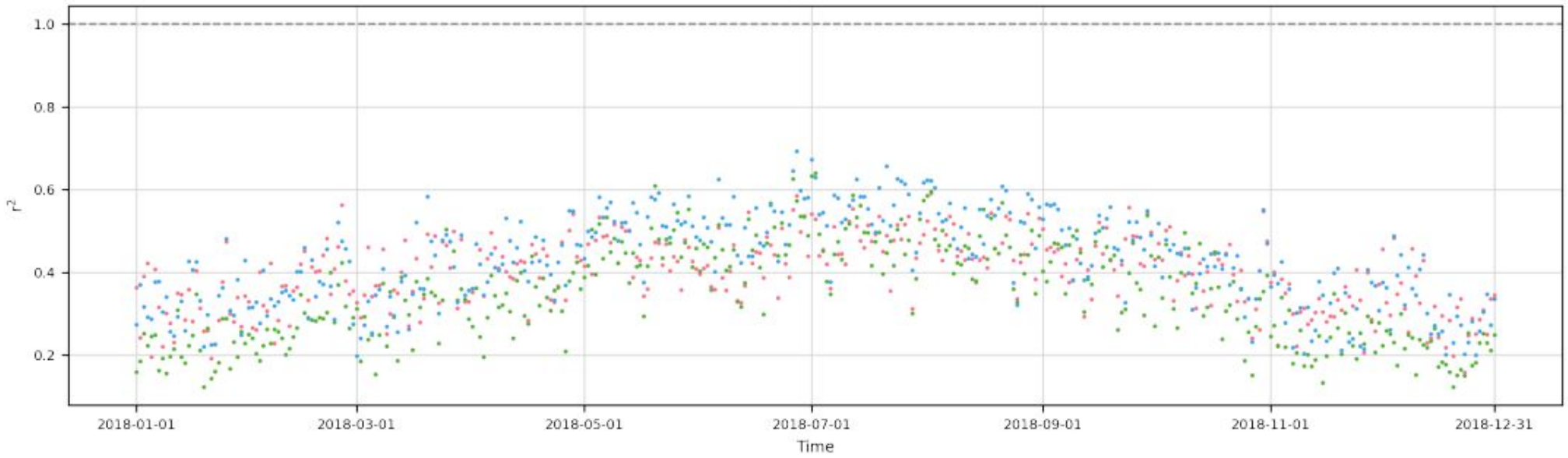
taylor-r



-[stat]-[resolution]

The timeseries can also be used to show how statistics vary in time. In order to do this, we need to add **-[stat]** and the temporal resolution after the plot type name (e.g. *timeseries-Mean-daily, timeseries-r2-monthly, timeseries-r-annual)*.

timeseries-r2-daily



Plot options

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Plot types

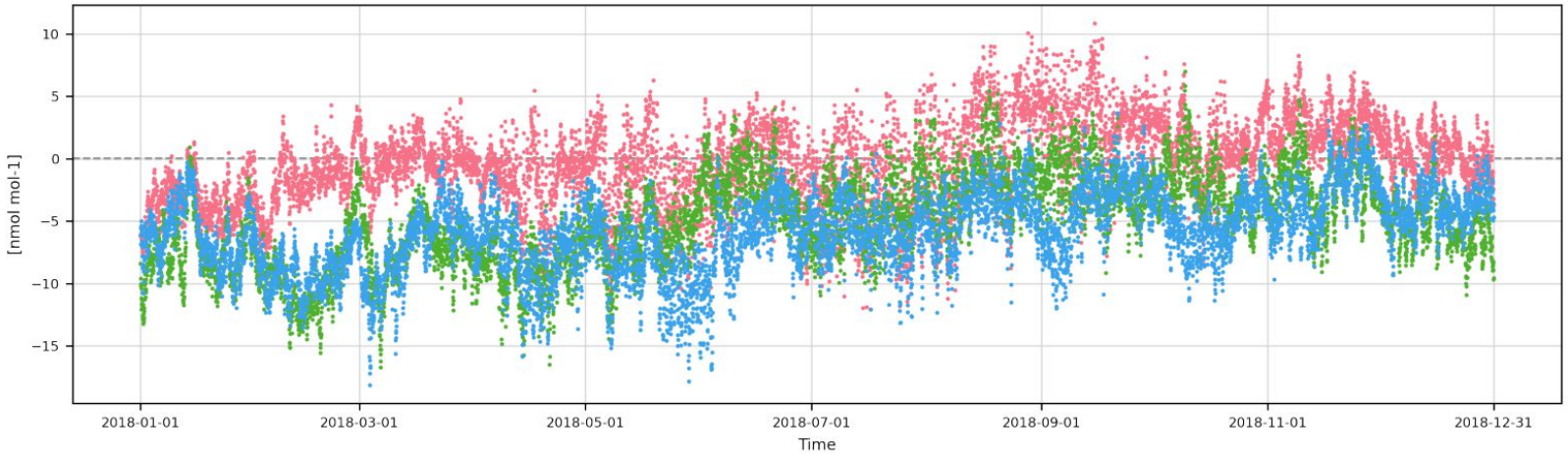
- Make bias (_bias)
- Show annotate (_annotate)
- Show label plots (_individual)
- Show observation plots (_obs)
- Make multispecies (_multispecies)
- Show logarithmic in x axis (_logx)
- Show logarithmic in y axis (_logy)
- Add regression line (_regression)
- Add smooth line (_smooth)
- Hide points (_hidedata)
- Show domain (_domain)
- Add threshold line (_threshold)
- Normalise y axis (_normalise)
- Show Gerrity scores (_gerrity)

Reference: https://providentia.readthedocs.io/en/latest/Plot-types-and-options.html

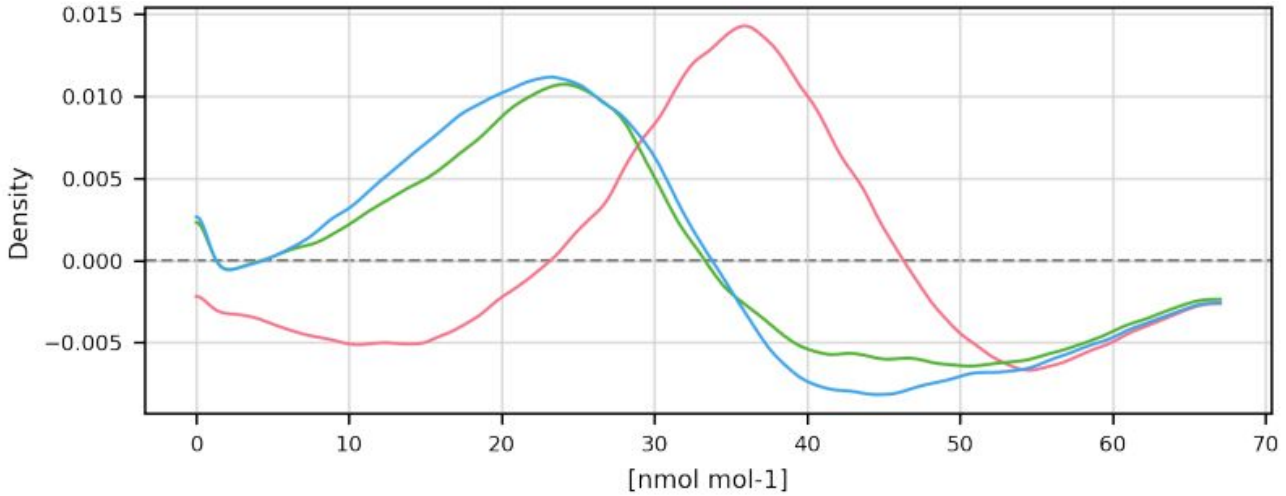
_bias

- **Dashboard**: timeseries, periodic, distribution, statsummary
- **Report**: map, timeseries, periodic, distribution, heatmap, table, statsummary
- **Library**: timeseries, periodic, distribution, heatmap, table, statsummary

timeseries_bias



distribution_bias



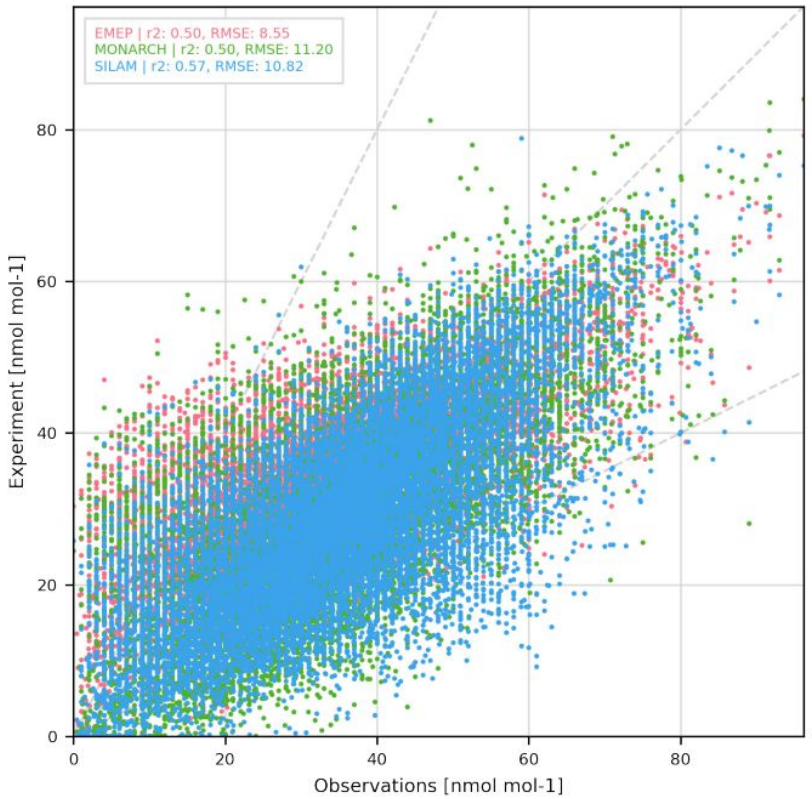
_annotate

- **Dashboard / Report / Library**: map, timeseries, periodic, periodic-violin, distribution, scatter, boxplot, taylor, fairmode-target

The statistics to annotate are defined per plot type in **settings/plot_characteristics.yaml.**

```
"scatter":
   {
    "grid": { "axis": "both", "color": "lightgrey", "alpha": 0.8 },
    "annotate_stats": ["r2", "RMSE"],
    "annotate_offset": { "loc": "upper left" },
    "annotate_bbox":
     {
       "facecolor": "white",
       "edgecolor": "gainsboro",
       "alpha": 1,
       "zorder": 100,
     },
     …
```

scatter_annotate



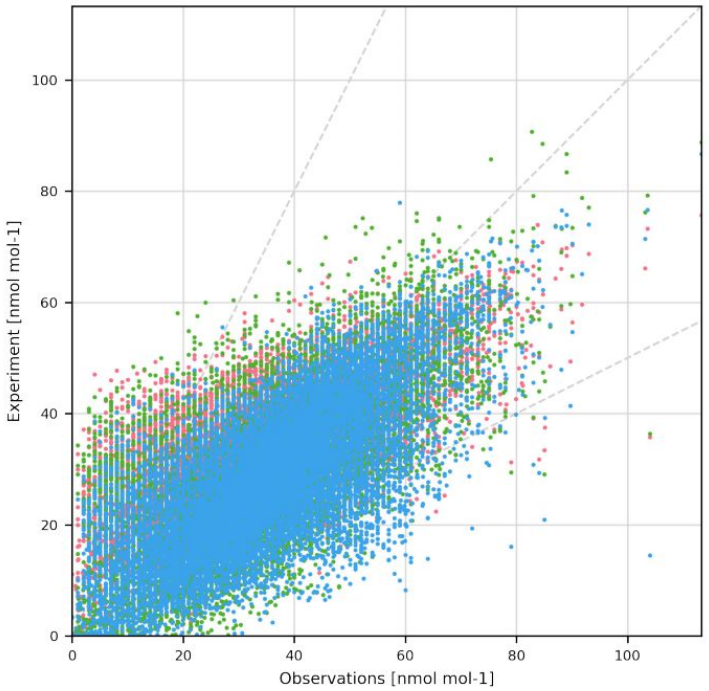
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

_individual

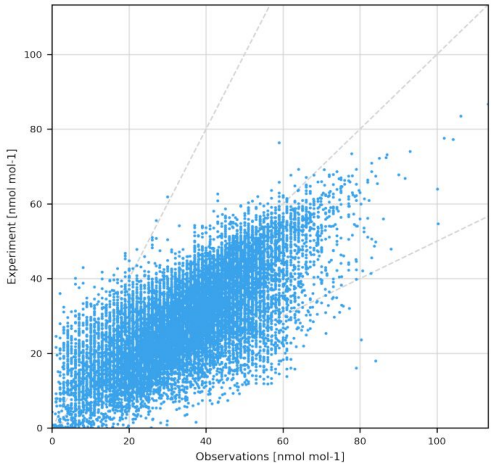
- **Dashboard / Library**: Not applicable
- **Report**: timeseries, periodic, periodic-violin, distribution, scatter, boxplot, taylor, fairmode-target, fairmode-statsummary

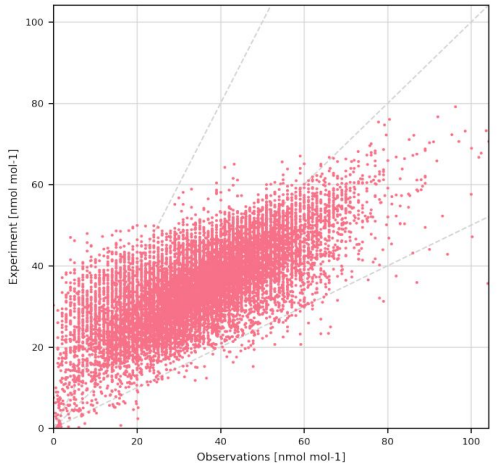
Sometimes you might want to show the data per model to see it with more detail.

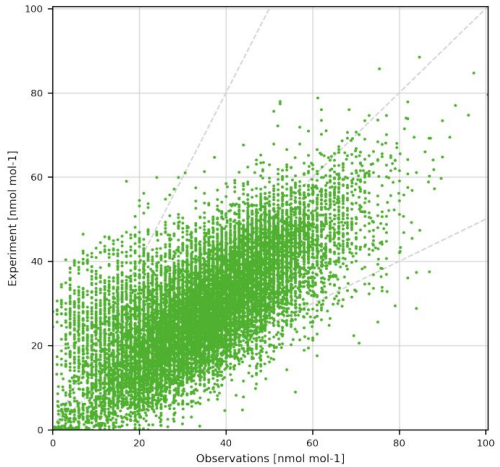
scatter



scatter_individual







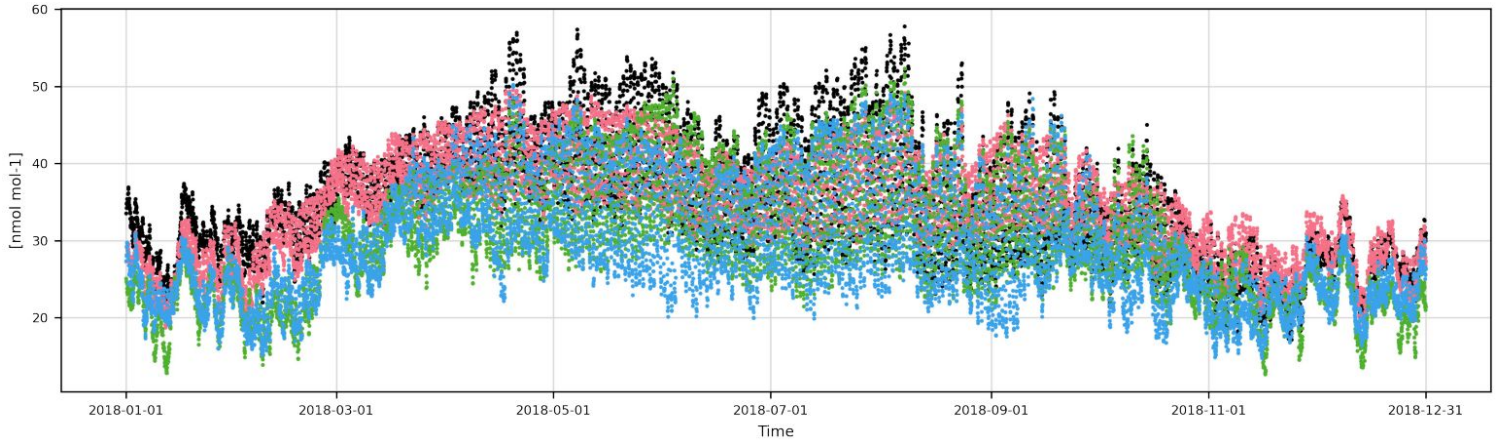
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

_obs

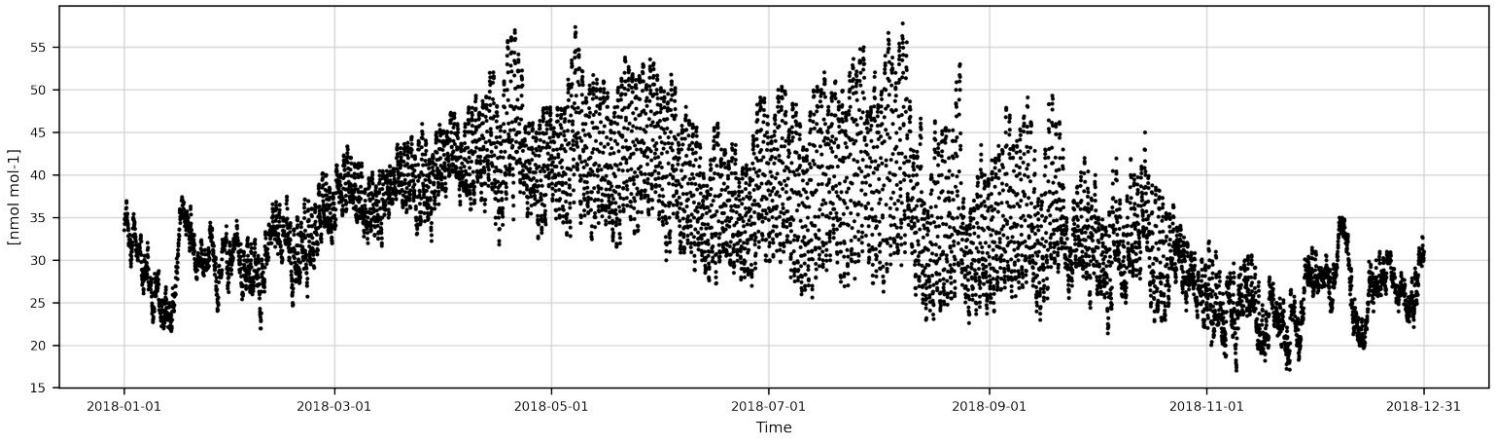
- **Dashboard / Library**: Not applicable
- **Report**: map, timeseries, periodic, periodic-violin, distribution, boxplot

To isolate the observations data

timeseries



timeseries_obs

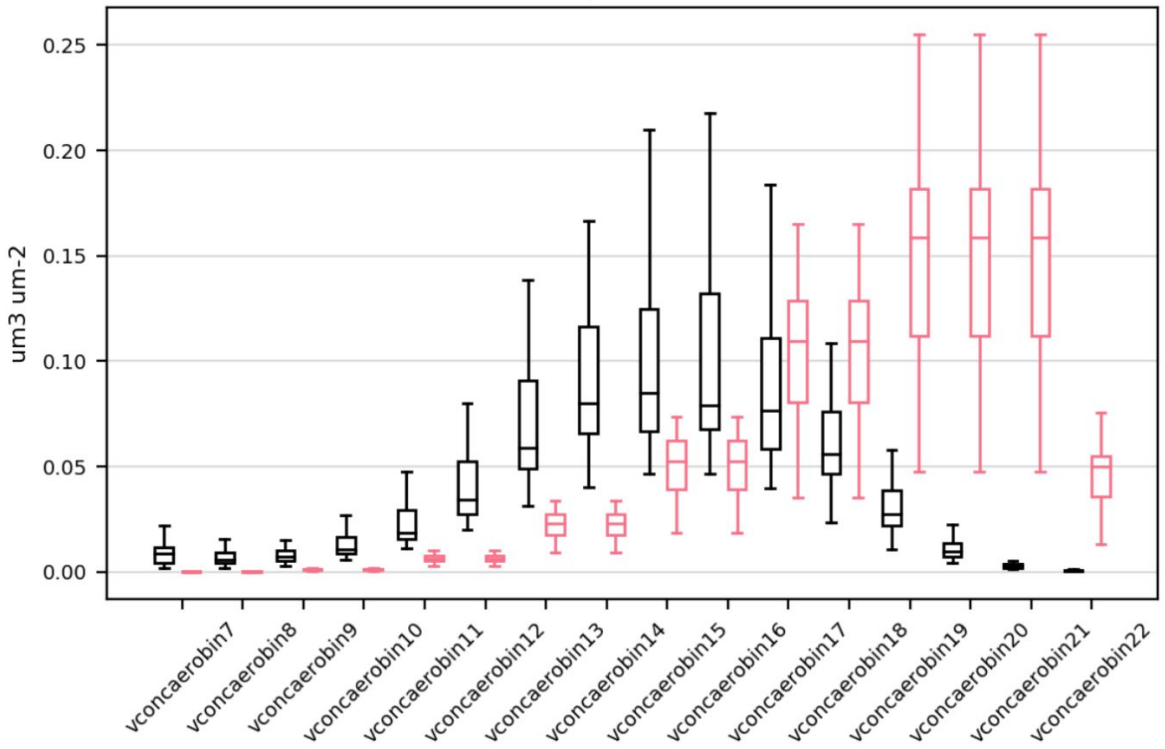


_multispecies

- **Dashboard**: Not applicable
- **Report / Library**: boxplot, heatmap, table, statsummary

The report mode is the only one able to plot multiple species per plot.

boxplot_multispecies



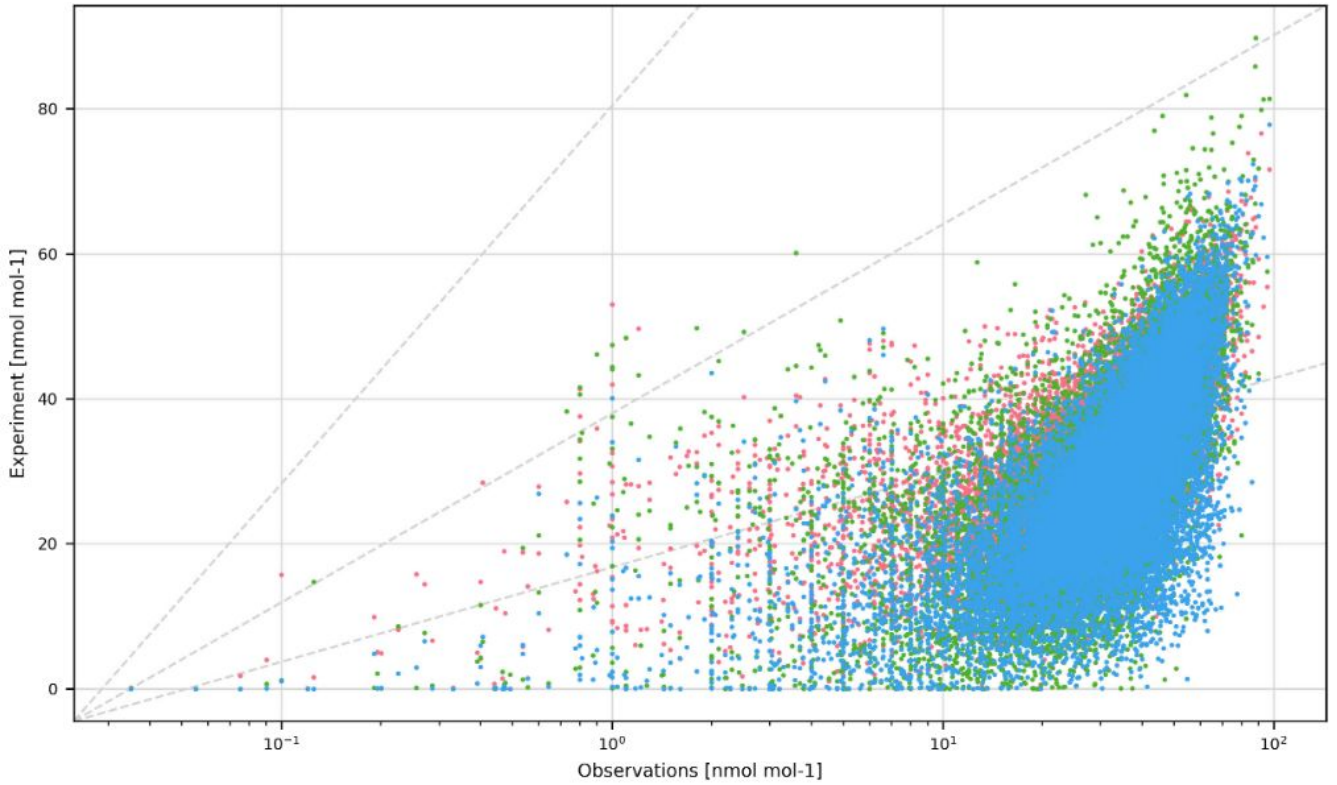
statsummary_multispecies

		p5	Mean	StdDev	p50	p95
sconco3	Observations	50.0	71.03	14.59	69.42	96.0
	Forecast	40.44	60.34	13.72	60.12	83.31
	Analysis	47.14	67.25	13.7	65.81	89.43
sconcno2	Observations	5.64	8.5	2.21	8.0	13.08
	Forecast	1.66	4.59	2.34	4.31	9.09
	Analysis	2.92	5.65	2.09	5.34	9.69
sconcco	Observations	0.2	0.24	0.02	0.24	0.28
	Forecast	0.13	0.15	0.01	0.14	0.17
	Analysis	0.16	0.18	0.01	0.18	0.2
sconcso2	Observations	2.0	2.1	0.16	2.0	2.49
	Forecast	0.45	0.81	0.24	0.82	1.2
	Analysis	0.95	1.45	0.29	1.52	1.82
pm10	Observations	12.01	15.91	2.43	16.19	19.46
	Forecast	5.99	8.44	1.6	8.17	10.96
	Analysis	10.43	13.91	2.27	14.06	17.3
pm2p5	Observations	5.7	8.83	1.84	9.02	11.55
	Forecast	4.18	6.17	1.33	6.03	8.52
	Analysis	5.12	8.15	1.72	8.29	10.84

- **Dashboard / Report / Library**: distribution, scatter

_logx

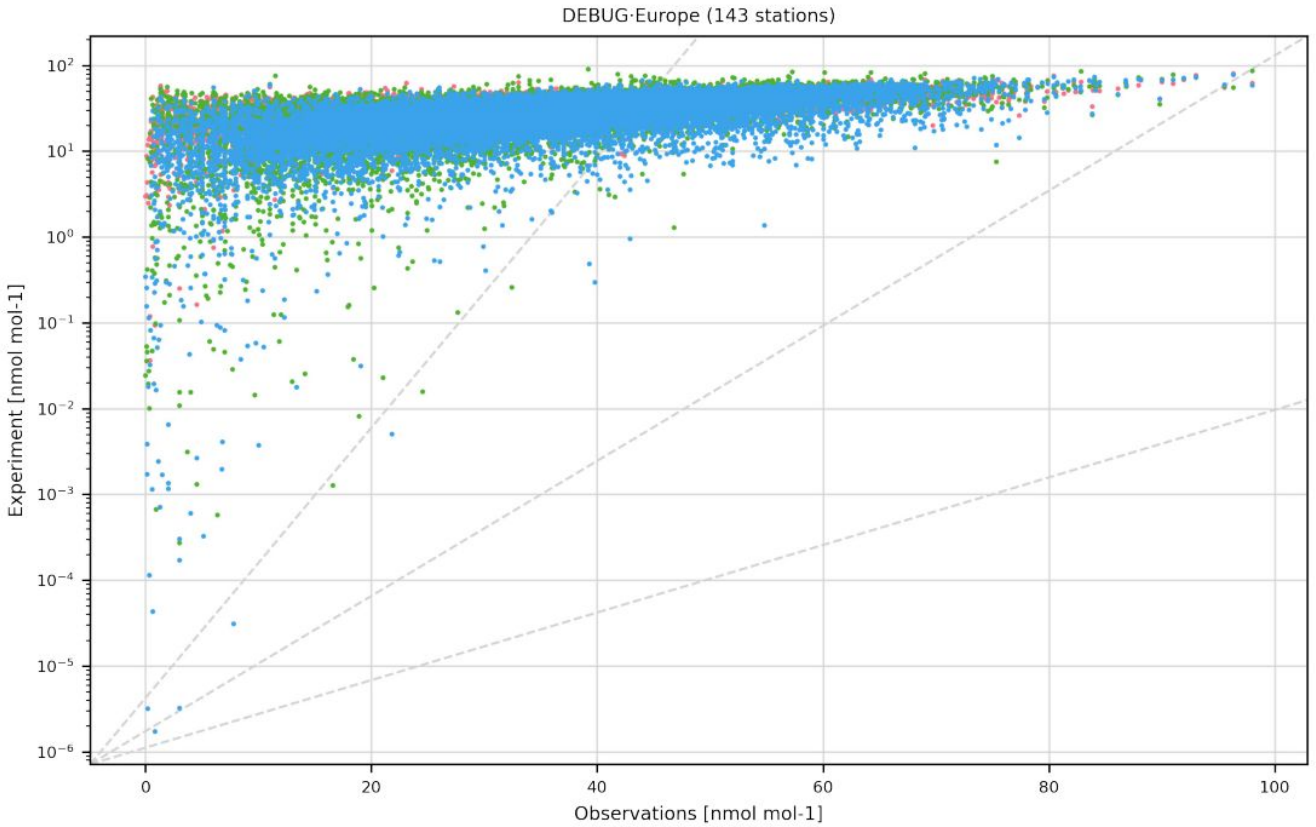
scatter_logx



_logy

- **Dashboard / Report / Library**: timeseries, periodic, periodic-violin, distribution, scatter, boxplot

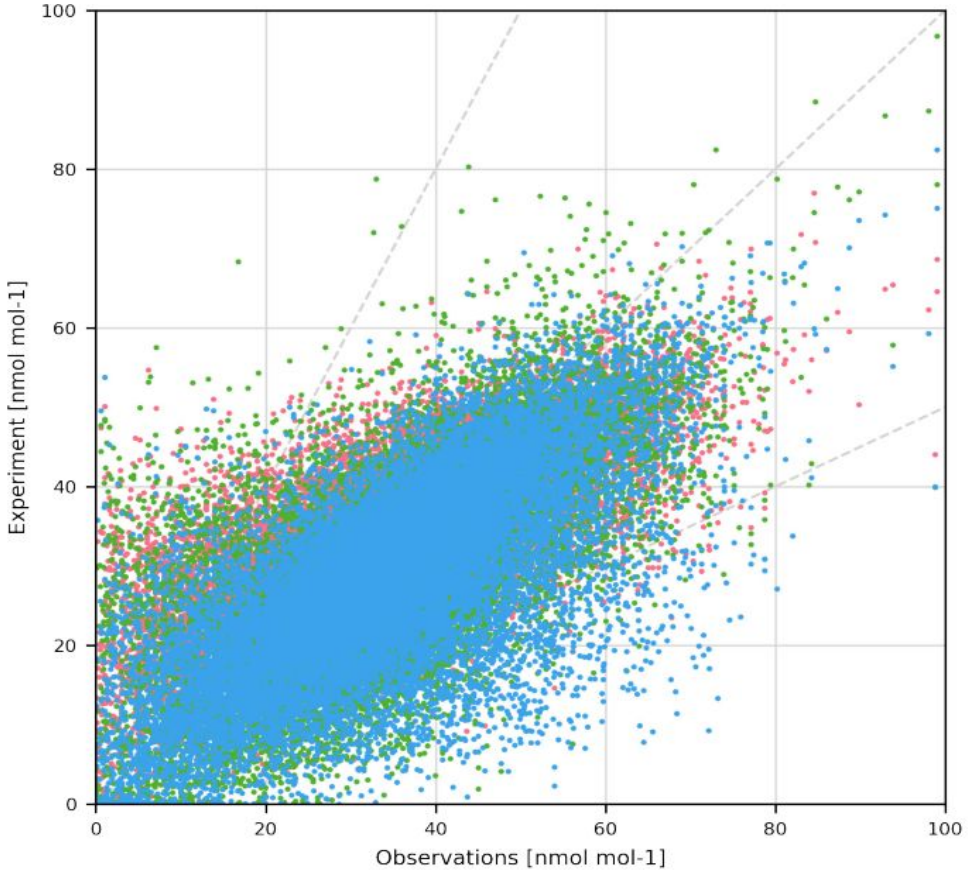
scatter_logy



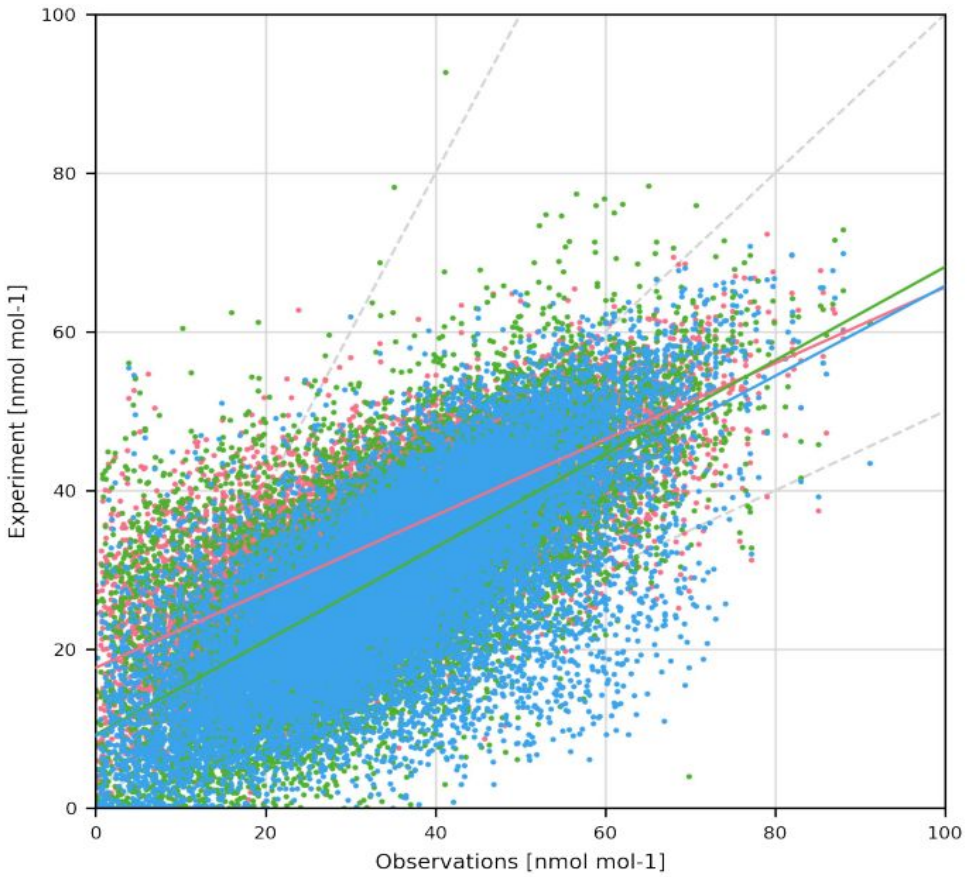
- **Dashboard / Report / Library**: scatter

_regression

scatter



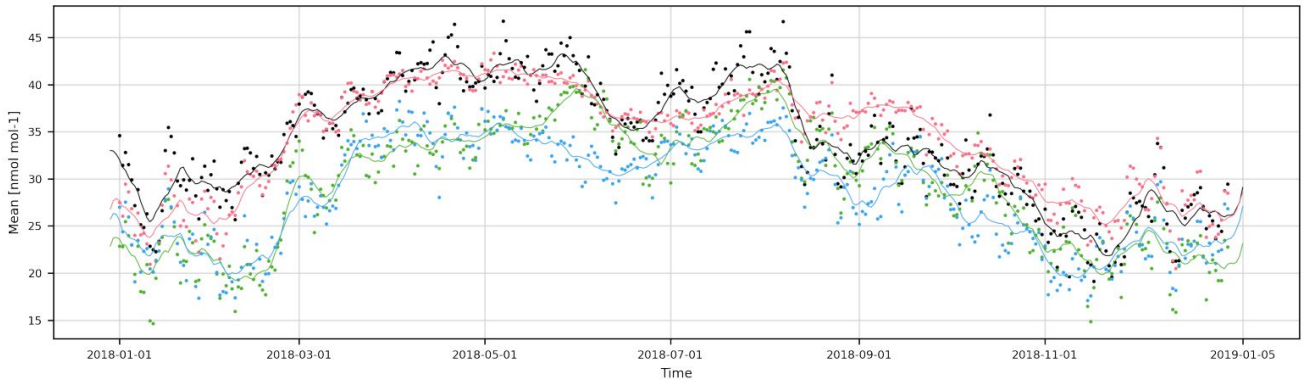
scatter_regression



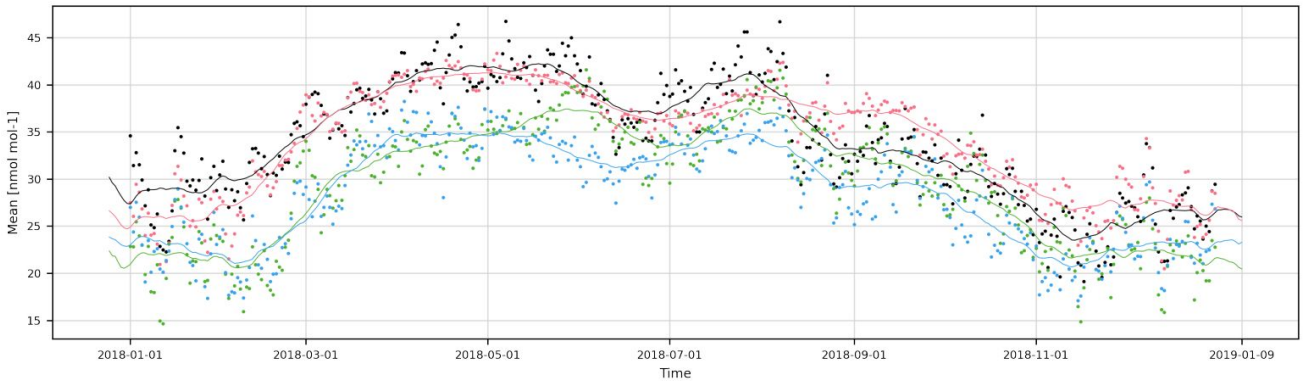
_smooth

The smooth line in the timeseries plot can be adjusted by setting the window and minimum points percentage in **settings/plot_characteristics.yaml.**

timeseries-Mean-daily_smooth (window=10)



timeseries-Mean-daily_smooth (window=30)

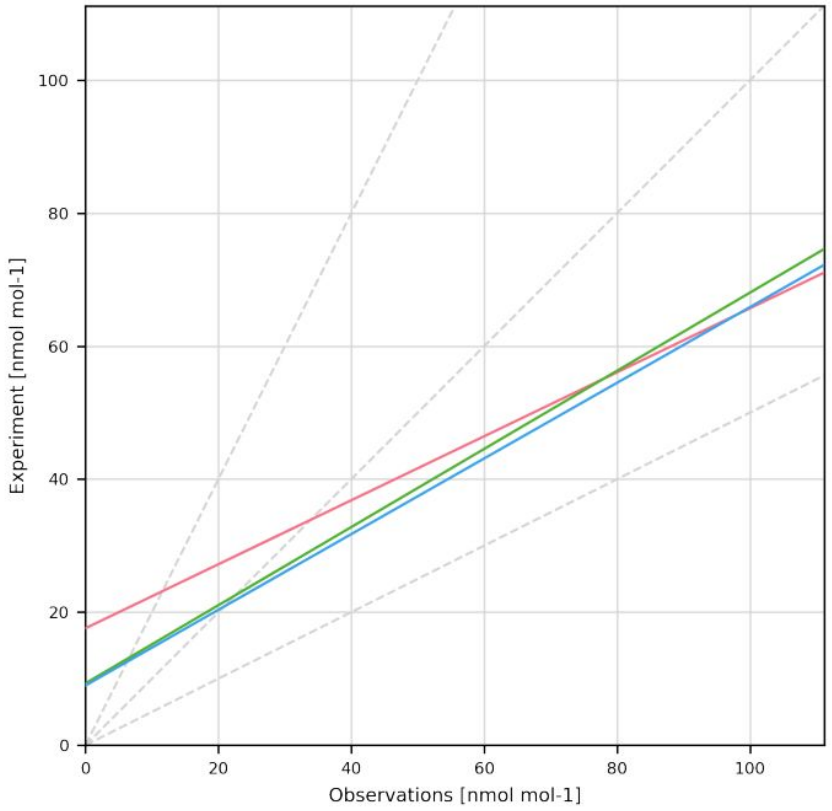


- **Dashboard / Report / Library**: timeseries, scatter

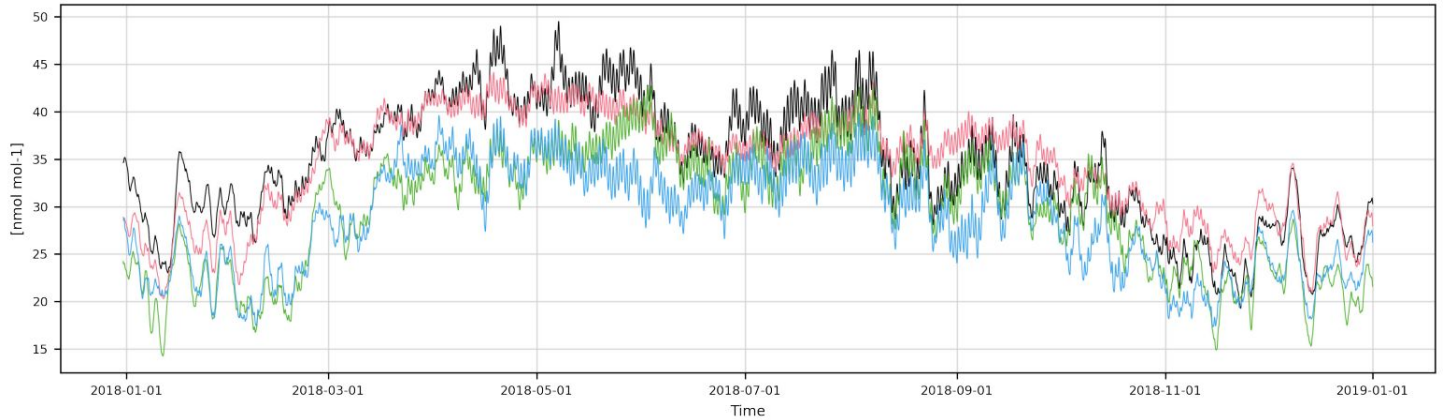
_hidedata

You can hide the points to only show the smooth or regression lines by using this option.

scatter_regression_hidedata



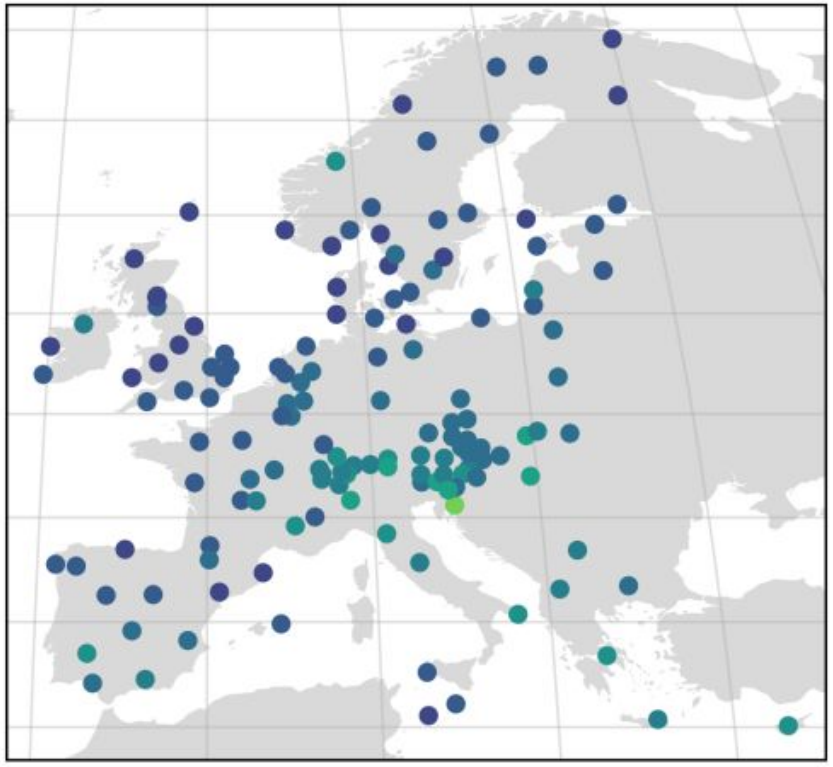
timeseries_smooth_hidedata



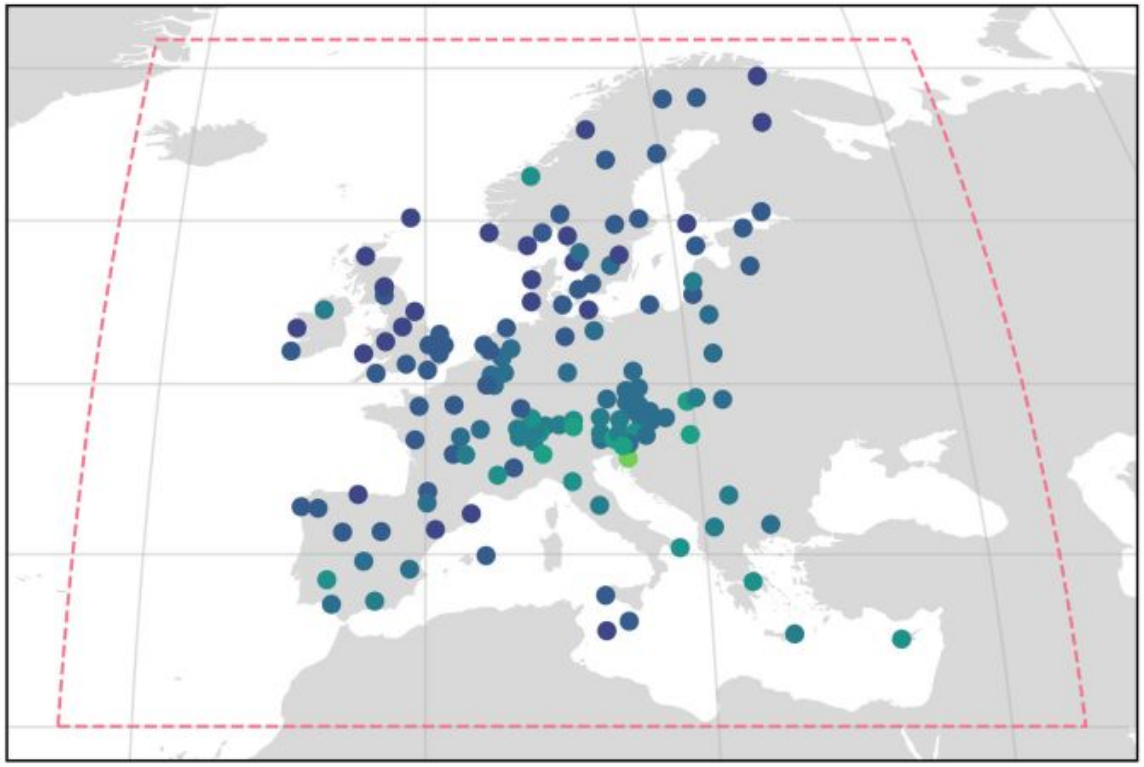
- **Dashboard / Report / Library**: map

_domain

map-RMSE



map-RMSE_domain

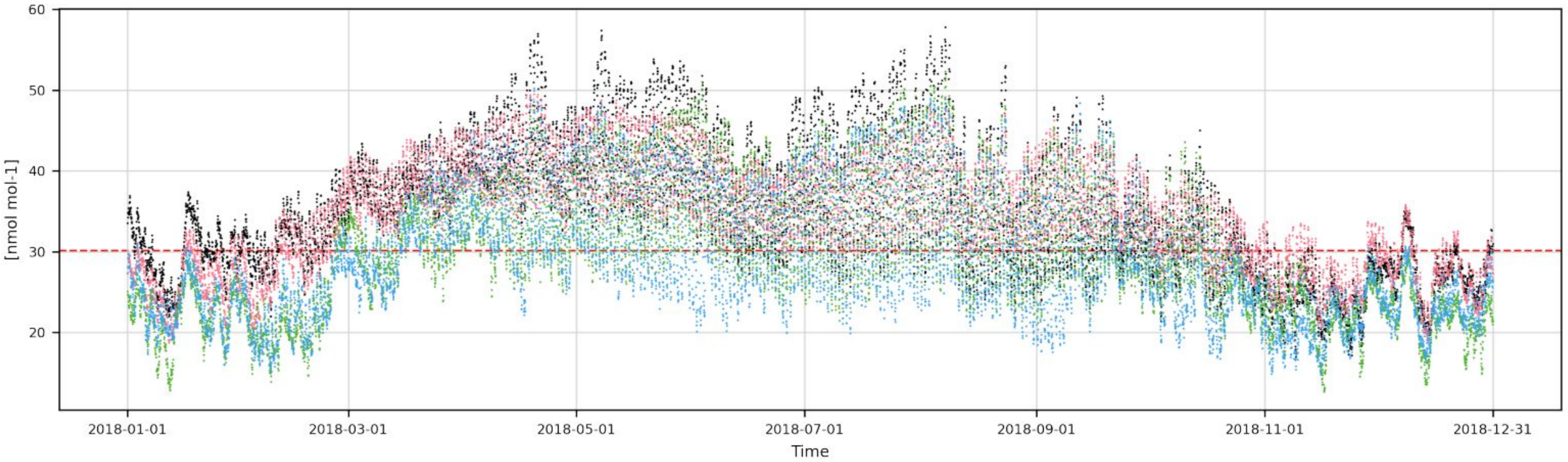


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

_threshold

- **Dashboard / Report / Library**: timeseries, periodic, periodic-violin, distribution, scatter, boxplot

timeseries_threshold (sconco3: 30.07)



_threshold

The **Exceedances** statistic gives the number of instances above a threshold. The threshold values can now be set in the file **settings/exceedances.yaml** per component, or network-component pair, as so:

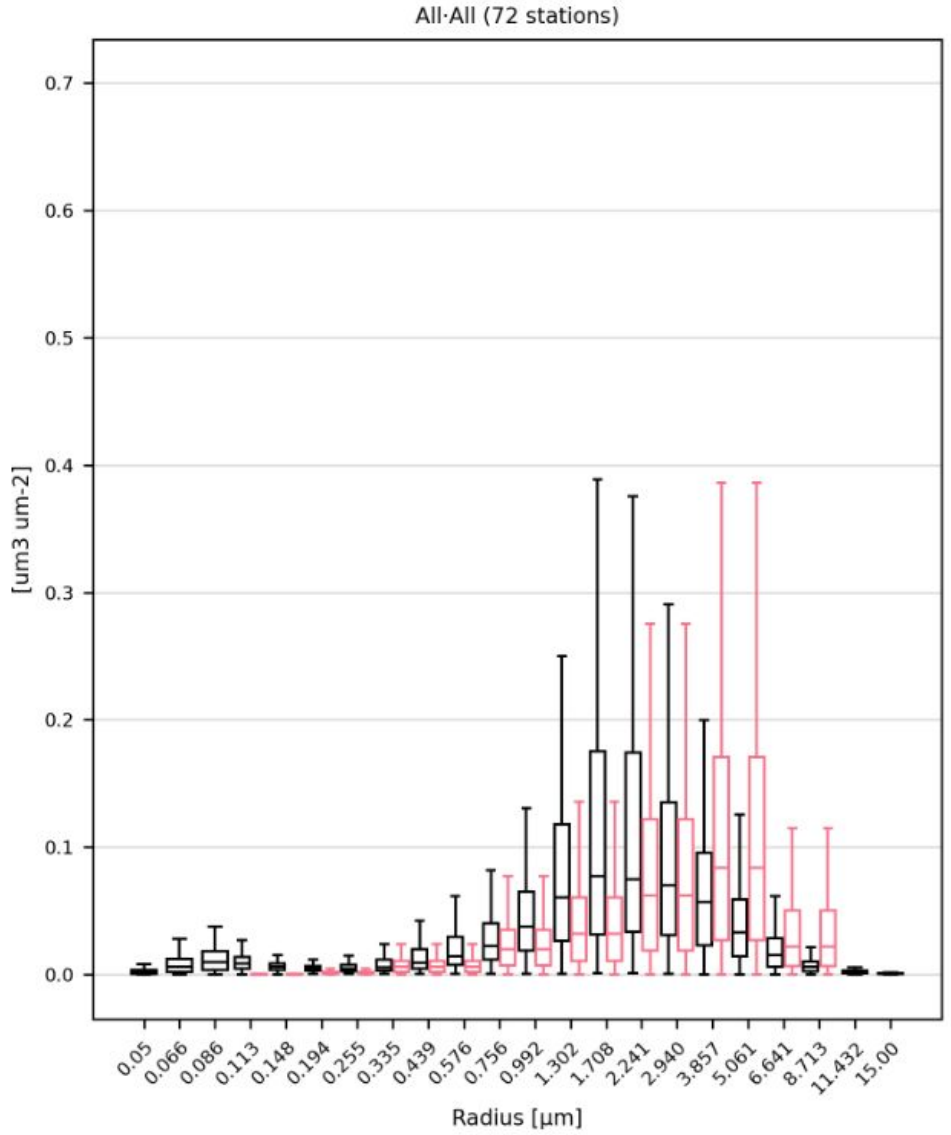
settings/exceedances.yaml

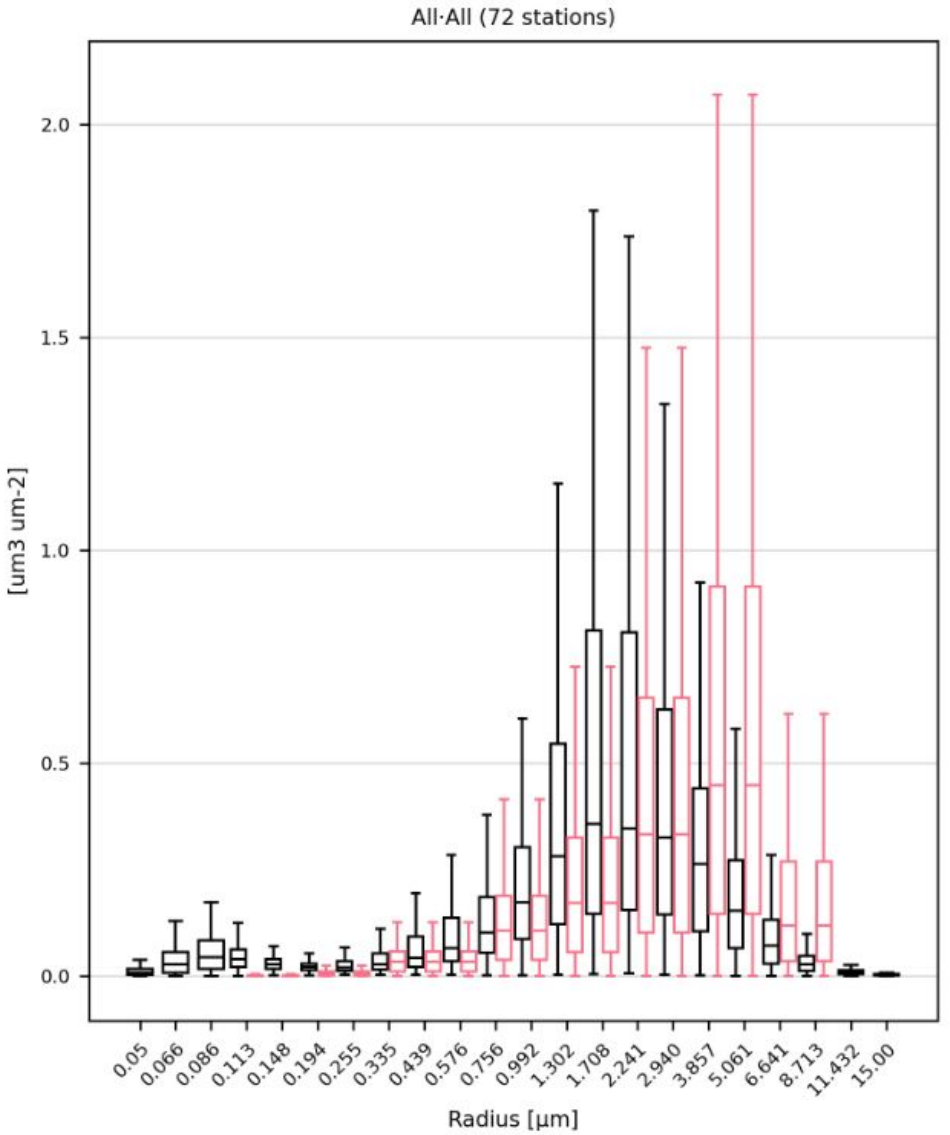
```
sconco3: 30.07
sconcno2: 5.23
sconcso2: 15.02
sconcco: 3435.18
pm2p5: 5.0
pm10: 15.0
```

In the case a threshold is set for a specific component, and per network-component, then the threshold for network-component is taken preferentially.

_normalise

- **Dashboard**: Not applicable
- **Report/Library**: boxplot





- **Dashboard/Report/Library**: contingencytable

_gerrity

Station reference	Station name	Latitude	Longitude	Gerrity Score
AT0002R_UVP	Illmitz	47.77	16.77	0.2
AT0005R_UVP	Vorhegg	46.68	12.97	0.06
AT0030R_UVP	Pillersdorf Bei Retz	48.72	15.94	0.17
AT0032R_UVP	Sulzberg	47.53	9.93	0.14
AT0034G_UVP	Sonnblick	47.05	12.96	0.18
AT0040R_UVP	Masenberg	47.35	15.88	0.14
AT0041R_UVP	Haunsberg	47.97	13.02	0.17
AT0042R_UVP	Heidenreichstein	48.88	15.05	0.14
AT0043R_UVP	Forsthof	48.11	15.92	0.16
AT0045R_UVP	Dunkelsteinerwald	48.37	15.55	0.17
AT0046R_UVP	Gänserndorf	48.33	16.73	0.2
AT0047R_UVP	Stixneusiedl	48.05	16.68	0.19
AT0048R_UVP	Zoebelboden	47.84	14.44	0.12
AT0049R_UVP	Grebenzen Bei St. Lamprecht	47.04	14.33	0.13
AT0050R_UVP	Graz Lustbuehel	47.07	15.49	0.13
DE0054R_UVP	Zugspitze-Schneefernerhaus	47.42	10.98	0.06

Report practical

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Report practical

Make 3 reports showing hourly data in January 2023 of the EEA network (GHOST version 1.6). Make a report that visualises:

1. The distribution of **rural** ozone using the a59j-regional-006 model in **Spain**.
2. The differences in ozone concentrations given by the a59g-regional-000 **forecast** in the **Mediterranean** and **Scandinavia**. Estimated coordinates:
   - Mediterranean: longitudes are -5 to 38, latitudes 32 to 43.
   - Scandinavia: longitudes are -1 to 32, latitudes from 55 to 72.
3. The change of ozone in rural areas from the a59j-regional-006 model during the month when nitrogen dioxide is **below 2 ppbv** at **altitudes under 1000m**.

Reference: https://providentia.readthedocs.io/en/latest/Configuration.html

Useful metadata fields

area_classification

GSFC_coastline_proximity

country

latitude

longitude

altitude

station_reference

map_extent

station_name

Exercise 1

The distribution of **rural** ozone using the a59j-regional-006 model in **Spain**.

```
[REPORT1]
ghost_version = 1.6
network = EEA
species = sconco3
resolution = hourly
start_date = 20230101
end_date = 20230201
model = a59j-regional-006
area_classification = keep: rural
country = keep: Spain
report_type = training
report_filename = Report1
```

Exercise 2

The differences in ozone concentrations given by the a59g-regional-000 **forecast** in the **Mediterranean** and **Scandinavia**. Estimated coordinates:

- Mediterranean: longitudes are -5 to 38, latitudes 32 to 43.
- Scandinavia: longitudes are -1 to 32, latitudes from 55 to 72.

```
[REPORT2]
ghost_version = 1.6
network = EEA
species = sconco3
resolution = hourly
start_date = 20230101
end_date = 20230201
model = a59g-regional-000
forecast = day
report_type = training2
report_filename = Report2

   [[Mediterranean]]
   longitude = -5, 38
   latitude = 32, 43
   map_extent = -6, 39, 32, 44

   [[Scandanavia]]
   longitude = -1, 32
   latitude = 55, 71
```

Exercise 3

The change of ozone in rural areas from the a59j-regional-006 model during the month when nitrogen dioxide is **below 2 ppbv** at **altitudes under 1000m**.

```
[REPORT3]
ghost_version = 1.6
network = EEA
species = sconco3
resolution = hourly
start_date = 20230101
end_date = 20230201
filter_species = EEA:sconcno2 (:, <2.0, nan)
model = a59j-regional-006
report_type = training3
altitude = 0, 1000
report_filename = Report3
area_classification = keep: rural
```

Library

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Library

The library functionality allows Providentia to be used as a Python module, **with access to its backend functions**, allowing different parts of Providentia to be used in scripts, as required. In that way, we can import it:

```
import providentia as prv
```

From a Python script or notebook outside Providentia it is important to add the path to your directory to the system paths before importing Providentia:

```
import sys
sys.path.append("path/to/your/Providentia/directory")
```

Examples: https://github.com/BSC-ES/providentia/tree/master/tutorials

Opening Jupyter notebook session

One typical use case is the interactive evaluation in a Jupyter notebook. Providentia has an in-built manner of opening a Jupyter notebook. If we add --notebook or --nb as a launch option in the command line, a Jupyter session will start:

```
./bin/providentia --notebook
```

Running locally, Jupyter **will open automatically**.

BSC HPC users:

If you are on a HPC machine this this **will launch an associated slurm job**, and a file will be created (notebook.out) containing the command to create an SSH tunnel on your local machine, which should be pasted into the terminal, and the path to open Jupyter in your browser.

Loading a configuration file

Once imported the first step is calling the class Providentia with the configuration you plan to use.

```
provi = prv.Providentia("interactive_template.conf")
```

This class' methods can be used:

- To access the modes: download, interpolate, report and dashboard.
- To load, filter, save and plot the data.

You can specify the section or subsection you want to work with by. Otherwise, if you have multiple sections, the first one will be selected unless it is explicitly defined.

```
provi = prv.Providentia("interactive_template.conf",
section="SECTIONNAME", subsection="SECTIONNAME·SUBSECTIONNAME")
```

Overwriting configuration parameters

If wanting to overwrite any overwrite any variables defined in the .conf file, each variable can be simply passed as an argument when loading the data, e.g.:

```
provi.load("interactive_template.conf", network="EANET")
```

These variables can be any variable that can be set in a .conf file.

Reference: https://providentia.readthedocs.io/en/latest/Configuration.html

Viewing active configuration

It is possible to view the active configuration variables in two different ways:

- Using the method *print_config*:

```
provi.print_config()
```

  This method can also be used to view another configuration file, by passing the filename:

```
provi.print_config("important.conf")
```

- By printing the active class instance:

```
print(provi)
```

Accessing the modes

The *download*, *interpolate, report* and *dashboard* functionalities can be integrated in workflows with little effort.

```
provi.download()
provi.interpolate()
provi.report()
provi.dashboard()
```

Loading the data

When the data is available in your machine, you can load it by doing:

```
provi.load()
```

Once the data has been loaded, it can be easily accessed by these methods:

- Retrieving a specific variable, by providing a variable name to the *variable* method.

```
var_data = provi.variable("network|component_variablename")
```

- Retrieving all metadata / data variables by using the *data* method. The returned data can be in different formats: nc (netCDF), np (numpy) or xr (xarray) by passing the format argument:

```
data = provi.data(format="xr")
```

Applying filters

To apply a filter not set in the configuration file, the *filter* method is used. The fields to filter by can be any data /metadata variable standardised in GHOST, and can be textual or numeric.

- If the field is numeric, use *lower* and/or *upper*, e.g.:

```
provi.filter(field, lower=28, upper=31)
```

- If the field is textual, use *keep* and/or *remove*, e.g.:

```
provi.filter("country", keep="Spain")
```

```
provi.filter("country", remove=["Spain", "France"])
```

- If the field is a representativity field, use *limit*:

```
provi.filter(rep_field, limit=20)
```

Selecting stations

To select station for one or more stations, you can use the function filter_station:

```
provi.filter_station(['AT0002R_UVP','SK0007R_UVP'])
```

Reset filters

If at any time any active data / metadata filters are wished to be reset, this can be done by using the *reset* method.

To reset all filters to the Providentia defaults, including those set from the original read configuration file:

```
provi.reset()
```

To keep the filters defined in the configuration file, but reset the ones added later, pass the *initialise* argument as True:

```
provi.reset(initialise=True)
```

Plotting

You can use the plotting functions of Providentia with the *plot* method. Any of the available plot types in Providentia can be made by passing the desired type.

```
provi.plot(plot_type)
```

Plots can also be saved to file, by passing *save* and a filename with the wanted extension (e.g. png, etc.):

```
provi.plot(plot_type, save="myplot.png")
```

The plotting object can also be returned for additional tinkering, by setting *return_plot* to True:

```
plot_obj = provi.plot(plot_type, return_plot=True)
```

Reference: https://providentia.readthedocs.io/en/latest/Plotting.html

Calculating statistics

All defined basic and model bias statistics defined in Providentia can be calculated from the loaded data, using the *statistic* method:

```
stat_calc = provi.statistic(stat, labela="OBS")
```

where *stat* is the statistic wished to be calculated, and *labela* is “observations”/model name or the alias set in the conf file. If wanting to calculate a bias statistic, you need to set *labelb:*

```
stat_calc = provi.statistic(stat, labela="OBS", labelb="EMEP")
```

For bias statistics for which a subtraction is involved, it is always done as *datab* - *dataa*.

If wanting to calculate statistics per station, *per_station* can be passed as True:

```
stat_calc = provi.statistic(stat, labela="OBS", labelb="EMEP", per_station=True)
```

Saving the data

Save the current configuration:

```
provi.save(format='conf')
```

Save the data as netCDF:

```
provi.save(format='nc')
```

Save the data as numpy:

```
provi.save(format='np')
```

If you want to save the data to a specific location, use the argument fname. By default, the data is saved in the folder **providentia/saved_data**:

```
provi.save(fname=''data/output/config.conf', format='conf')
```

Library practical

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Library practical

Launch the notebook mode, create a notebook and load the training configuration file. Use **hourly** data and **GHOST version 1.6**.

1. Create a **map** showing the **median** of the a59j-regional-006 model for EEA **ozone** stations in **Spain**.
2. Create a map showing the **RMSE** of the **rural nitrogen dioxide** stations using the a59j-regional-006 model in **Germany**.
3. Read the data to find out the available **station references** and select **one particular station** to create a **timeseries** with **statistical annotations** of the ozone concentrations.
4. Reset the filter and show a **multispecies statsummary bias** and **heatmap** showing the **r** statistic in the ozone concentrations given by the **two models**.
5. Save the heatmap as an **image** and the data as a **netCDF** file.

Customisation

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Plot customisation

To update the style of plots in the dashboard, reports and library, edit the plot characteristics in **settings/plot_characteristics.yaml**.

settings/plot_characteristics.yaml

All

Dashboard

Report

Library

```
"timeseries":
 {
  "grid": { "axis": "both", "color": "lightgrey", "alpha": 0.8 },
  "threshold_line":
   { "linestyle": "--", "linewidth": 1.0, "color": "red", "zorder": 0 },
  …
  "dashboard":
   {"remove_spines": ["top", "right"], ...},
  "report":
   {"orientation": "landscape", ...},
  "library":
   {"figsize": [17, 6], ...},}
```

Reference: https://providentia.readthedocs.io/en/latest/Plot-customisation.html

Colorbar customisation - Bounds

There are two ways to change the colorbar bounds:

- If you want to set the same bounds for all statistics, you can edit the parameters *vmin_absolute*, *vmax_absolute*, *vmin_bias* and *vmax_bias* under *map* in the plot characteristics files.
- If you want to set the bounds for each statistic (recommended), you should edit the same parameters in **settings/basic_stats.yaml** and **settings/model_bias_stats.yaml**.

Colorbar customisation - Color

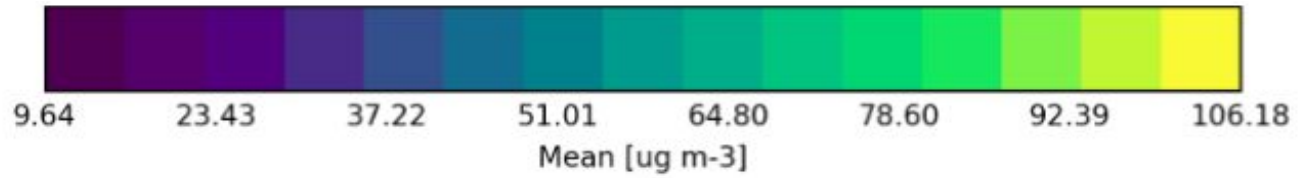
To change the colors, you need to edit the cmap:

- If you want to set the same colors for all statistics, edit the parameters *cmap_absolute* and *cmap_bias* under *map* in the plot characteristics files.
- If you want to set the colors for each statistic (recommended), you should edit them in **settings/basic_stats.yaml** and **settings/model_bias_stats.yaml**.

Colorbar customisation - Discretisation

You might also want to change the number of breaks if you have a discrete colorbar. For this you will need to change the number of ticks (*n_ticks*) and number of discrete colors (*n_discrete*) under *map* in the plot characteristics files.

In this example, we have set *n_ticks* to be 8 and *n_discrete* to be 15.



If you prefer to have a continuous colorbar and want to remove the breaks, you need to set *discrete* to be false.

Colorbar customisation - Per statistic

Users can define the color and bounds of the colorbar per species using a dictionary.

settings/basic_stats.yaml

```
"Mean": {"function": "calculate_mean",
         "order": 0,
         "label": "Mean",
         "arguments": {},
         "units": "[measurement_units]",
         "minimum_bias": [0.0],
         "vmin_absolute": {"sconco3": 0, "sconcno2": 0},
         "vmax_absolute": {"sconco3": 20, "sconcno2": 5},
         "vmin_bias": {},
         "vmax_bias": {},
         "n_ticks": {},
         "n_discrete": {},
         "cmap_absolute": {"sconco3": "viridis", "sconcno2": "viridis"},
         "cmap_bias": "RdYlBu_r"}
```

Legend customisation

If you want to change the color of the legend, you can edit the *legend_color_palette* under *general* in the plot characteristics files.

The default palette is *husl*:

Legend and colorbar customisation - Color

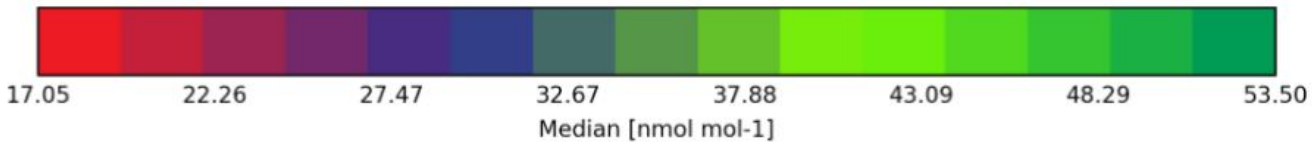
Now the legend and colorbar colormaps can be customized by adding a new schema in **settings/color_palettes.yaml**.

```
"cams": ["#ED1B23", "#2D2F92", "#7CFC00", "#009B55"]
```

To update the legend colors, you will need to set the custom name in *legend_color_palette* in the plot characteristics files.

For the colorbar, you should define *cmap_absolute* or *cmap_bias*, depending on the statistic type.

The colors will interpolate depending on n_discrete. Below n_discrete is 15.



Labels customisation

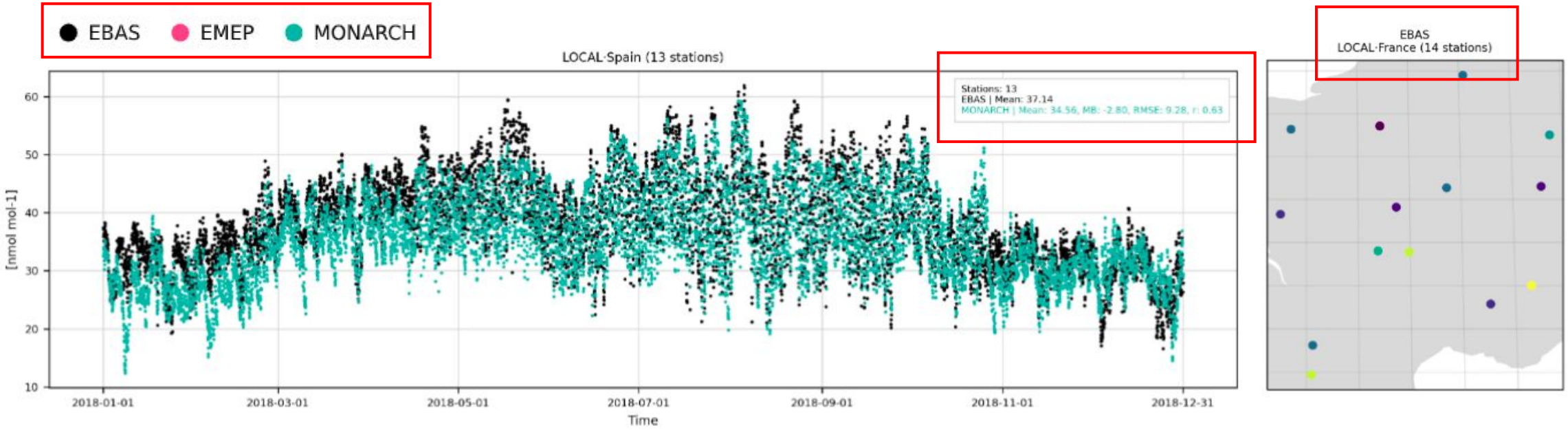
By defining **observations_data_label** in our configuration files, we can customise the name of the observations in the legend, plots and maps.

observations_data_label = EBAS

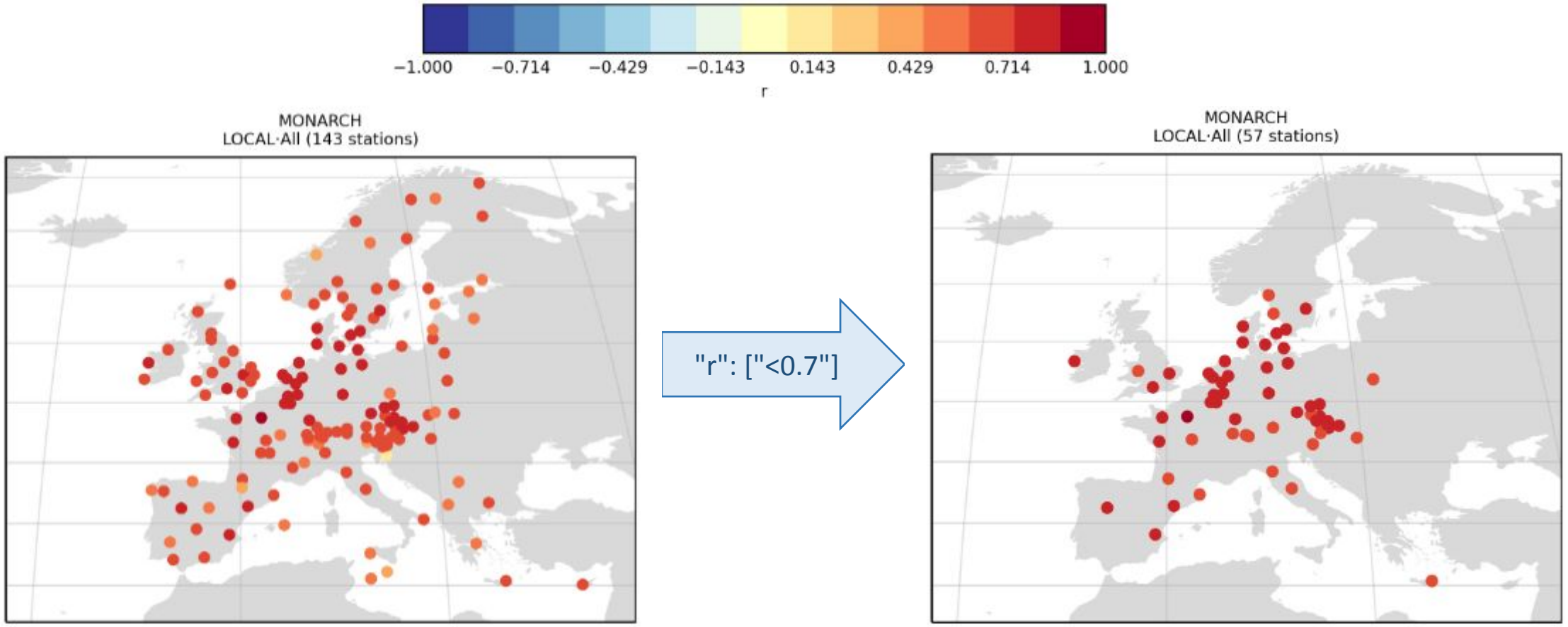
If you want to edit the names of the models, you can only do it launching Providentia with a configuration file, where you will define the alternative names as in:

model = cams61_emep_ph2-eu-000, cams61_monarch_ph2-eu-000 (EMEP, MONARCH)

Labels customisation



Remove stations by statistics



Remove stations by statistics

If you want to automatically remove stations that have certain statistical values, you will need to add your criteria in the file **settings/remove_extreme_stations.yaml**. An example of this exists for CAMS:

configurations/configuration_name.conf

```
[CAMS2_40]
network = ineris/eionet-cams2_40-ira
species = sconco3,sconcno2,sconcco,sconcso2,pm10,pm2p5
resolution = hourly
start_date = 20230101
end_date = 20230215
model = a59g-regional-000, a59j-regional-006 (Forecast,
Analysis)
temporal_colocation = True
spatial_colocation = False
report_type = operational
report_summary = True
report_stations = False
report_title = CAMS2_40 Forecast and Analysis Report
report_filename = operational_report
statistic_mode = Temporal|Spatial
statistic_aggregation = Median
periodic_statistic_mode = Independent
periodic_statistic_aggregation = Mean
remove_extreme_stations = CAMS
```

settings/remove_extreme_stations.yaml

```
"CAMS": {"r": ["<0.3"],
         "NMB": ["<-100.0", ">100.0"],
         "NRMSE": [">100.0"]}
```

Remove stations by statistics

Any absolute statistic can be set to be a bias statistic by adding _bias e.g.:

"p95_bias": ["<10",">20"]

The statistics can be general, across all components, or they can be specific per component, for example:

settings/remove_extreme_stations.yaml

```
"CAMS": {"r": {"sconco3": ["<0.3"],
                "sconcno2":[<0.55]},
          "NMB": {"sconco3": ["<-100.0", ">100.0"],
                  "sconcno2": ["<-20.0", ">20.0"]},
          "NRMSE": {"sconco3": [">100.0"],
                    "sconcno2":[">200.0"]}}
```

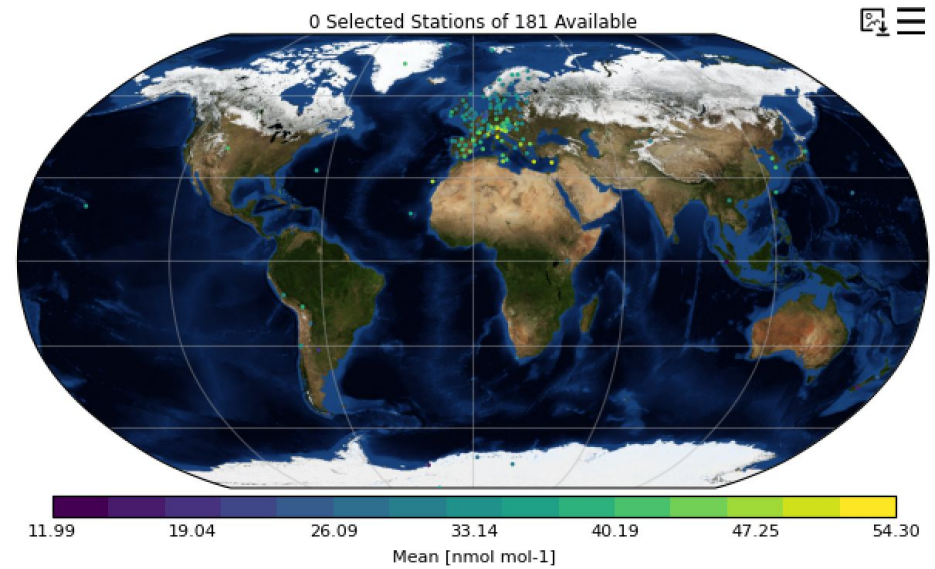
Map background

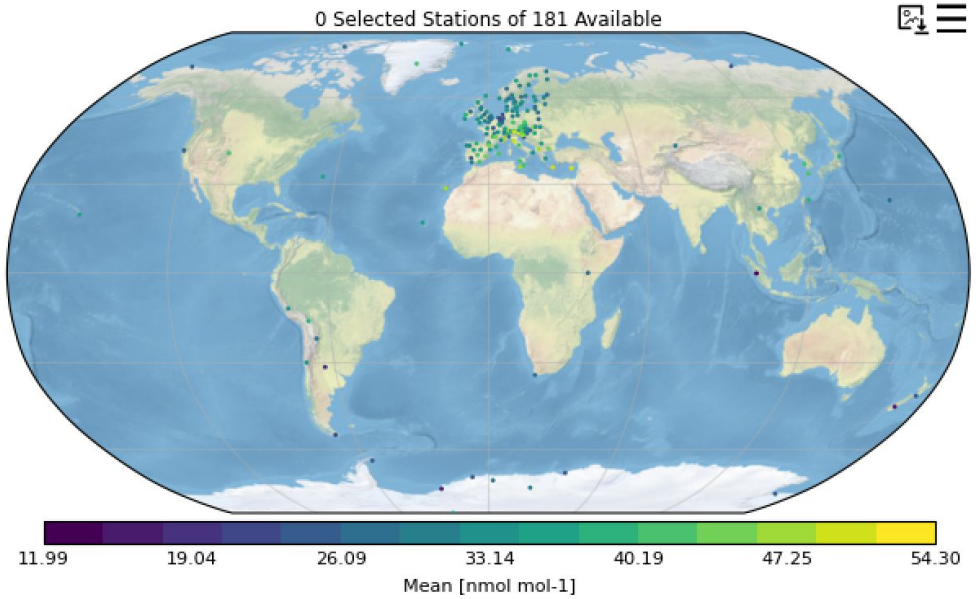
Users have the ability to plot a background image on the map axis by changing the ***background*** variable under the ***map*** section in the **settings/plot characteristics.yaml**.

There are 3 available standard options:

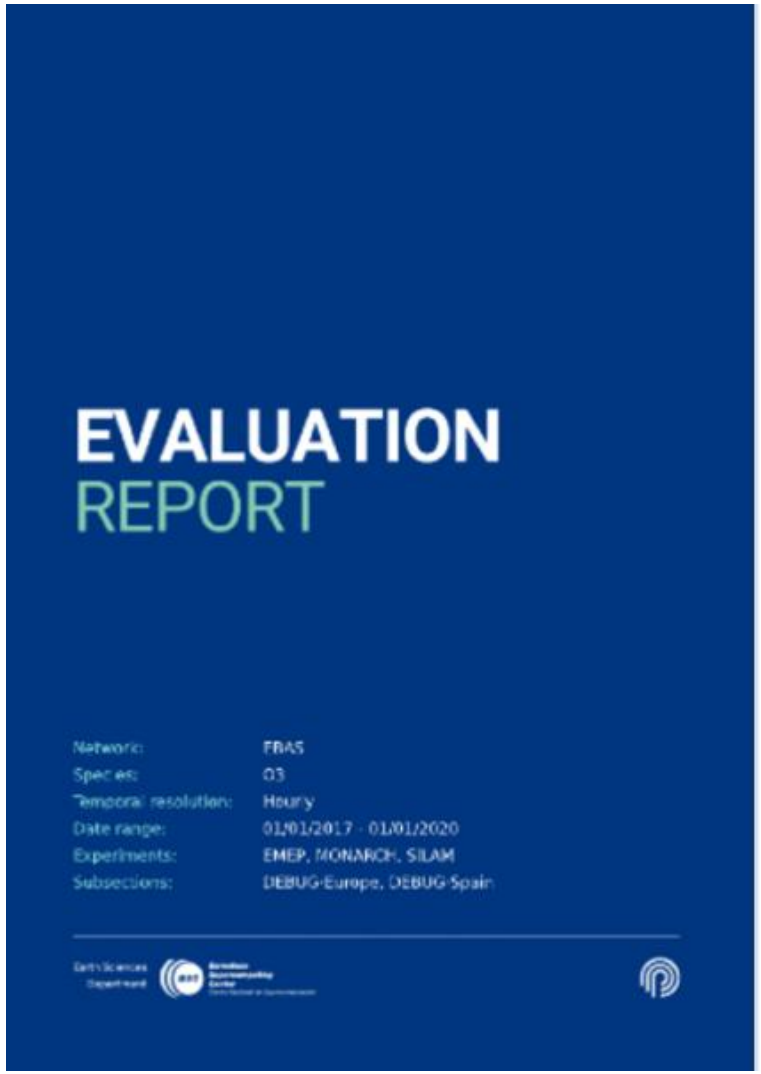
1. ***providentia*** (default)
2. ***blue_marble*** (NASA's blue marble product)
3. ***shaded_relief*** (imagery showing changes in evolution)

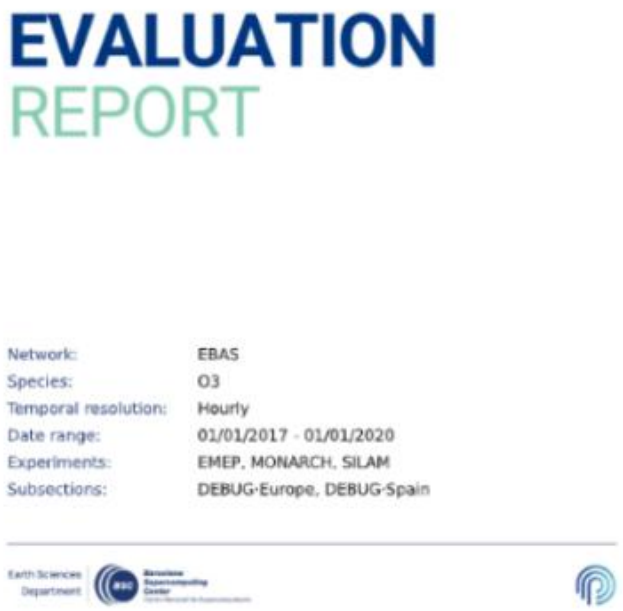
However, any image can be set. Check the Plot Customisation wiki page for all the details.





Report cover page



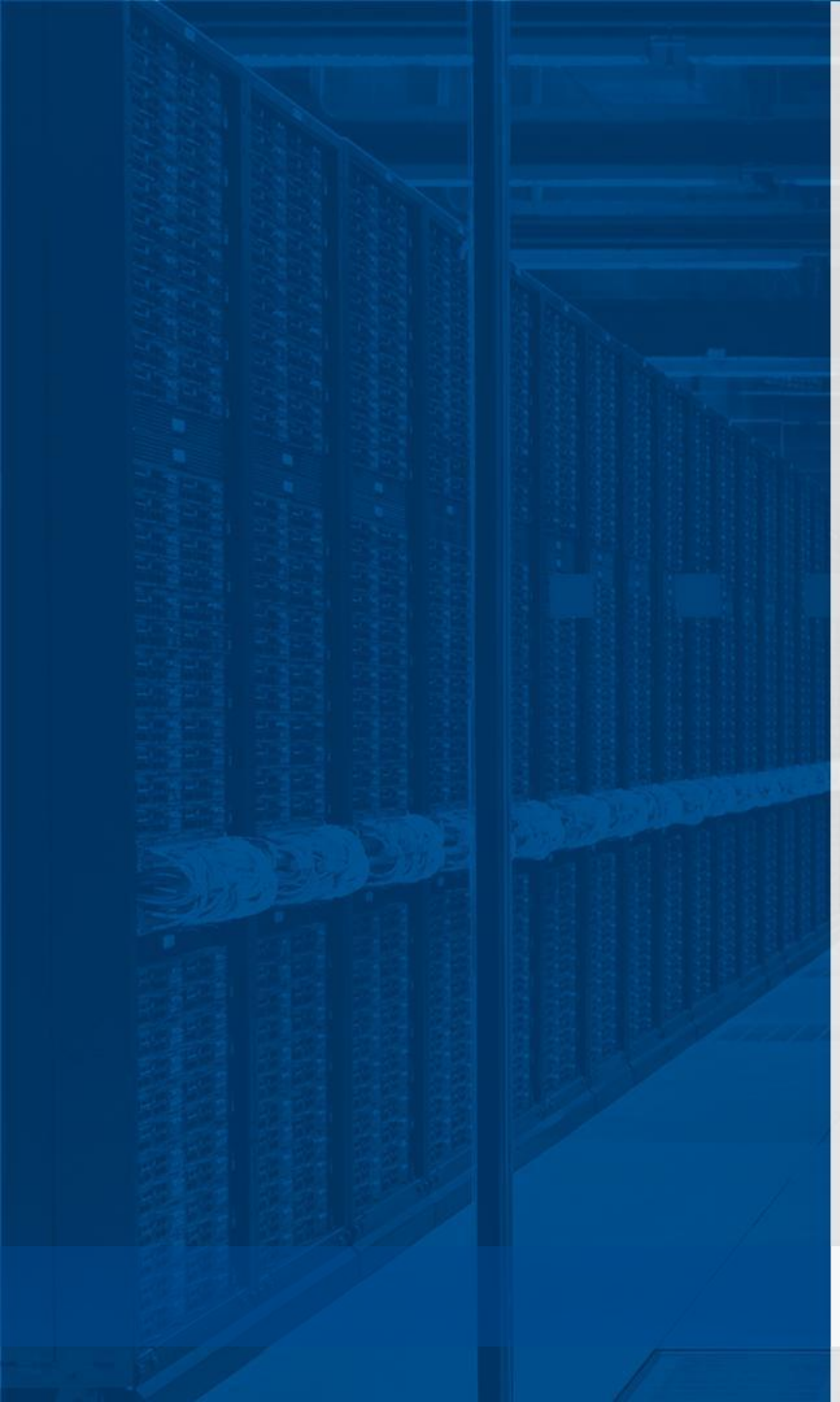


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Report cover page

The cover page can be customised by editing the parameters under **header** in **settings/plot_characteristics.yaml** file. The most interesting ones are these:

- **dark_mode** to set the background to be dark (blue tone) or light (white).
- **variables** to specify which variables you want to show. The options are network, species, resolution, dates, model, temporal_colocation, spatial_colocation, filter_species, calibration and subsections. By specifying the value under those keys, you can overwrite the default variable values and write anything you want.
- **logo** to display any logo on your report cover. To use it, you must specify the path to your PNG file for the corresponding background (dark or light mode).



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Thank you for your attention!

More information at:
https://github.com/bsc-es/providentia

Join the #providentia Slack channel!

paula.serrano@bsc.es | alba.vilanova@bsc.es | dene.bowdalo@bsc.es